THE UNIVERSITY OF SALFORD

THE DEVELOPMENT OF AN INTELLIGENT INVENTORY MANAGEMENT SYSTEM

YANSONG LIANG

TIME
RESEARCH INSTITUTE

DEPARTMENT OF COMPUTER AND MATHEMATICAL SCIENCES, UNIVERSITY OF SALFORD, SALFORD, UK

Submitted in Partial Fulfilment for the Degree of Doctor of Philosophy, April, 1997

CONTENTS

Chapter 1: Introduction

Chapter 2: Literature Review

Chapter 3: Overview of the Proposed Inventory ES

Chapter 4: Demand Pattern Identification

Chapter 5: Inventory Model Base Embedded in the Proposed System

Chapter 6: Development of the Intelligent Inventory System

Chapter 8: Conclusions and Further Work

Appendix 1: MANUAL FOR OPERATING THE SYSTEM

Appendix 2: THE DERIVATION OF SOME MODELS FOR CONSTANT DEMAND

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS USED

AI/ES	Artificial Intelligence/Expert system
AI	Artificial Intelligence
CBR	Case-Based Reasoning
CMA	Centred Moving Average
DDL	Dynamic Data Exchange
DEC	Digital Equipment Corporation
DLL	Dynamic-Link Libraries
DSS	Decision Support Systems
EDP	Electronic Data Processing
EOF	End of File
EOQ	Economic Order Quantity
ES	Expert Systems
IT	Information Technology
IDSS	Intelligent Decision Support Systems
JIT	Just-In-Time
KBS	Knowledge-Based Systems
MDI	Multiple-Document Interface
MIS	Management Information System
MRP	Material Requirements Planning
MS/OR	Management Science/Operational Research
OEL	Object Linking and Embedding
OR	Operational Research
ROI	Return on Investment
TRC	Total Relevant Cost
TRCUT	Total Relevant Cost per Unit Time
NTRC	Normalised Total Relevant Cost
SQA	Software Quality Assurance
V&V	Verification and Validation
UGI	Graphical User Interface

ACKNOWLEDGEMENTS

I would like to express my great thanks to my supervisor, Dr. Khariy Kobbacy for his excellent guidance, constructive discussions and constant encouragement.

My special thanks are also addressed to Mr. Robert Greenwood (Optical Fibres) and Ms. Jenny Turner (British Airways) for their kind support and co-operation.

The financial support of the Chinese government and the British Council are also gratefully appreciated.

Finally, my last thank you, but not the least is for my wife Hongyan and my daughter Helen who have been very patient and supportive.

DECLARATION

The software described and attached to this thesis can not be installed for commercial purpose without the prior permission of the author.

ABSTRACT

This thesis is concerned with the development of an intelligent inventory management system. The aim of the system is to bridge the substantial gap between the theory and the practice of inventory management and to help industrial inventory managers to achieve an effective and successful inventory management. The proposed system attempts to achieve this by providing automatic pattern identification and model selection facilities. Such a hybrid knowledge-based inventory system consists of a collection of techniques (or pattern identifier) for identifying demand and lead time patterns and a knowledge base (or rule base) for subsequent selection of a suitable inventory model taking into consideration aspects of the practical situation. There are no previous attempts in the inventory literature to develop such a system to guide model selection.

In order to integrate the system into the established computer-based intelligent inventory management system and facilitate the function of the pattern identifier, a data manager has been developed to manipulate the history data required for statistical analysis and to load the data into the system from other applications. In order to establish the system's model base, the study of the modelling of inventory and the features and evolution of expert systems are reviewed. The published models which deal with similar inventory problems have been compared based on its applicability, clarity, and being suitable to be computerised. It was necessary to further develop and amend published models to fill gaps in the model base.

The overall structure and salient features of the proposed system and the development of the system using Visual Basic have been described. The system has been tested using real life data supplied by the co-operating companies. Finally, achievements and shortcomings of the system are discussed and some suggestions for further research are outlined.

Chapter 1

INTRODUCTION

This chapter briefly describes the background and motivation for the research work described in this thesis. In section 1.1 the elements of inventory management is presented. This is followed in section 1.2 by discussing the substantial gap between the theoretical research and the practical applications which is the vital challenging problem encountered. A possible approach to solving this problem which motivated this work is then introduced in section 1.3. The chapter concludes with an outline for the focus and organisation of the thesis.

1.1 The Elements of Inventory Management

1.1.1 Introduction

Inventory is defined as a stock of items kept on hand by an organisation to use in meeting customers demand [1].

The importance of inventory to a firm stems from two points of view: financial and operational. First, inventory represents a major financial investment for any company. Inventories represent of 25 to 50 per cent of total assets in manufacturing firms and 75 to 80 per cent in wholesalers and retailers [2]. On the other hand, from the operational perspective, inventories add an operating flexibility. Adequate inventories kept in manufacturing companies will smooth the production process. The wholesales and retailers can offer good customer services and gain good public image by holding sufficient inventories. The basic objective of inventory management is to achieve a balance between the low inventory and high return on investment (ROI).

1.1.2 The Functions of Inventory

The functions that inventory performs can be summarised as follows [3]:

First of all, the fundamental function for carrying inventories is to meet customer demand for a product. In fact, it is physically impossible and economically impractical for each stock item to arrive exactly where it is needed exactly when it is needed. Therefore, a reasonable level of inventory is normally maintained that will meet anticipated or expected customer or user demand.

Secondly, since demand is usually not known with certainty, additional amounts of inventory, called safety or buffer stocks, are often kept on hand to meet unexpected variations in excess of expected demand.

Thirdly, additional stocks of inventories are sometimes built up to meet demand that is seasonal or cyclical in nature. Companies will produce items when demand is low in order to meet high seasonal demand for which their production capacity is insufficient. Correspondingly, retailers might find it necessary to keep large stocks of inventory on their shelves to meet peak seasonal demand, or for display purposes to attract buyers.

Finally, inventory can also be carried out to take the advantage of price changes. A company will often purchase large amounts of inventory to take advantage of price discounts, as a hedge against anticipated price increases in the future, or because they can get a lower price by purchasing in volume.

1.1.3 Classes of Inventory Problems

The crucial distinction in inventory management is drawn by the nature of demand for items. In 1965 a very useful classification of demand was proposed by Orlicky [4]. He used the term "independent demand" to describe any demand of items that is influenced by market conditions and unrelated to demand for other items in a company's inventory. This includes the demand for finished goods and spare parts. He also used the term "dependent demand" to describe any demand for items directly determined by other associated items. Typical of this are raw materials, purchased or manufactured parts or ingredients, and manufactured subassemblies, attachments and accessories.

It is Orlicky's classification that provides the real key to selection and applicability of inventory control techniques. Dependent demand, by definition, can be precisely determined from the demand of related items, and the methods of material requirements planning (MRP) and just-in time (JIT) are the appropriate techniques to treat this kind of inventory problem encountered in manufacturing companies [5, 6]. The key ingredients of MRP are master production schedule, bills of materials, and inventory recorders. Using information from these sources, the MRP system identifies actions such as releasing new production orders, adjusting order quantities, and expediting late orders. JIT systems are designed to produce or deliver goods or services as need, using minimal inventories. It is actually a philosophy that focuses on reducing inefficiencies and unproductive time in the production process. Both MRP and JIT are more than inventory control systems; they also involve process design and scheduling issues.

Table 1.1 Classification Scheme for Independent Demand Inventory Problems

Characteristic	Attributes
Number of items	One or many
Demand	Constant, stochastic,; static or dynamic
Lead time	Constant or stochastic
Stocking Location	One or many
Stockouts	Backorders or lost sales

For independent demand items, which are only considered in this work, there is a large variety of inventory problems identified by the essential characteristics of the inventory system (see Table 1.1) and many hundreds of inventory models have been developed to handle this kind of inventory problems.

1.1.4 Inventory Decisions

There are three primary decisions that must be made in regard to independent demand inventories if the managers attempt to accomplish the functions of inventory reviewed above.

1. How to monitor the inventory.
2. How much should be ordered.
3. When should orders be placed.

1.1.5 Inventory Modelling

The purpose of inventory modelling is to derive an operating doctrine. The methodology for modelling inventory situations is straightforward and three simple steps are involved [7]:

1. Examine the inventory situation carefully, listing characteristics and assumptions concerning the situation.

2. Develop the total annual relevant cost equation.

3. Optimise the cost equation to find the optimum for how much to order (order quantity) and when to reorder (reorder point).

1.1.6 Costs in Inventory

Most of the inventory models are built around the assumption that the objective is to minimise inventory costs that basically fall into four categories. In this thesis, the inventory costs are assumed to be constant.

1. Item cost is the cost of buying and/or producing the individual items. The cost of items is often an important consideration when quantity discounts are offered. The item cost can usually be obtained from vendors.

2. Ordering cost is incurred because of the work involved in placing purchase orders with vendors or to organise for production within a plant. This cost should include the costs of acquiring the data necessary for making decisions, computational cost, stationary, telephone calls, transportation, receiving and inspection. The ordering cost can be estimated from the company's records. However, difficulties are sometimes encountered in separating the fixed and the variable ordering-cost components.

3. Holding cost is associated with keeping items in inventory for a period of time. The holding cost usually consists of electricity and heat, insurance and tax, spoilage and obsolescence, the cost of capital and the expenses of running the warehouse. This cost is more difficult to determine accurately in terms of historical information [8]. In practice, however, the estimation of holding cost is often based substantially on managerial judgement.

4. Stockout cost reflects the economic consequences of running out of stock. It is the most difficult of all inventory costs to estimate. One approach is simply to specify an acceptable stockout risk level. Another interesting method used by Reimans et al. [9] is to treat the unit shortage cost as a function of the gross profit of an item. This approach has the advantage of resulting in better service for higher profit items.

It is relatively easy to list the contents of each category of inventory costs as mentioned above. However, their measurement in practice is a very difficult task. In particular, accounting information, primarily collected and recorded for a financial purpose, is usually inappropriate for estimating inventory costs. Furthermore, shortage costs are often not shown in accounting records. There is no satisfactory solution to this problem and more research work is needed to establish methods for estimating inventory costs.

1.2 Problem Statement

Since the first inventory model was published in 1915, many papers about independent demand inventory modelling have been published [10]. These provide a theoretical foundation for the field of inventory management and make the research in inventory one of the most developed fields of OR. However, the practical implementation of inventory models is much more behind the development of inventory modelling [11].

The discrepancy between theory and practice of inventory may come from both the academic and practical areas [12]. Much of the research is aimed at rigorous analysis of underlying equations representing the inventory problems and developing mathematical decision models. This type of theoretical work is most highly valued by the academic community. Therefore, there is often less attention given to providing workable solutions to real problems. On the other hand, inventory managers may not be aware of the mechanics and applicability of the theoretical models. This will also hinder the practical application of inventory models because an understanding of the fundamental structure of complex models is the first step necessary to provide a workable solution of the problem being considered.

Developments in information technology (IT) over the past two decades have enabled many organisations to establish computer-based information management system (MIS) to improve inventory management. Most of these systems are mainly used to record transactions, produce management reports, and monitor inventory status. They lack the ability to help inventory managers to choose the correct model by analysing the data kept by the system to identify the inventory environment.

To maintain optimal inventory operations in a company, the management will have to continuously review and update the accumulated data, the selection of suitable models and the computation of new optimum values of ordering decisions. This is needed because the conditions in relation to the demand, the cost elements, the supply, the lead time, etc., are likely to change with time. The complexity of such a problem will greatly increase as a result of a large number of stock items in the inventory. Many large industrial companies will stock thousands of items (see Section 7.2) which results in a very large number of records being generated. The analysis of such huge amounts of information is beyond manual human capacity or traditional computational methods.

For a company to be more competitive, it has to face the challenge of reducing the processing costs. This issue is closely related to the effectiveness of inventory management which can help to reduce both the storage and the labour operating expenses. A small percentage reduction in inventory can be transformed into significant operating profit.

1.3 A Proposed Solution to the Problem

The mathematical techniques and other methods are only aids to management decision-making. They cannot replace the judgement of human experts. A manager may have several inventory models available to him, but if he is not sure which is the best one for the situation, obviously, he may not be able to solve the problem effectively. It is clear that when a manager is faced with an inventory problem and he has an expert available for choosing the right model then he can be confident that the efforts put into analysis will be efficient and successful. In recent years, the emergence of expert systems, described in Section 2.3, as a branch of artificial intelligence (AI) has provided a useful approach to solve this problem.

The approach espoused here is to harness computer power through the construction of a decision support system (DSS). Since the expertise of statistical forecasting and inventory modelling must be incorporated in the proposed system, so the DSS might be classified a knowledge-based or intelligent decision support system, The belief that a useful approach to address this problem might be offered by the techniques of computer based intelligent decision support system (IDSS), discussed in Section 2.2 and 2.4, led to the work described in this thesis. The aim of this research is to explore this approach by developing a prototype intelligent inventory management system.

The purpose of the intelligent decision support system, known as the proposed Intelligent Inventory Management System, is to provide advice to inventory managers to achieve an efficient and effective inventory management practice. The scope for using such computerised system in support of inventory managers' decision making would then be investigated by using real data from already-established computerised inventory database. The proposed system is designed to extract data from inventory database of the co-operating companies, analyse the data to specify the demand patterns and lead time distributions, use this information to select an appropriate inventory model for that particular environment, calculate the optimal order quantity, update the inventory status of the item, and present output to the user in both numerical and graphical format.

This system that embodies the published knowledge and expertise in the fields of the forecasting and inventory modelling into its knowledge base can provide inventory managers with mathematical modelling support. Such a system can also enable the rapid update of the analysis of the demand patterns and the lead time distributions as more data is collected. If the outcome resulting from the analysis indicates that the environment of an inventory item has changed, the system would automatically choose a new suitable model for the item given its present situation.

Further intelligent facilities would be desirable in such an IDSS, including two-way query between the user and the system. The user may wish to request justification of a recommendation given by the system, or the investigation of alternative solutions. Conversely, the system could be given the ability to query the user for judgements or for instructions. By beginning to make use of more aspects of the system and demonstrating that it can be of direct use in improving inventory management, it is hoped that inventory personnel may be encouraged to record more detailed operational data.

The effort presented in this thesis is more like attempting to automate the application of standard statistical tests in identifying inventory demand patterns and hence the selection and application of published inventory models. This work does not extend to use the expert experience on how to deal with case where standard methods fail to give a clear result

1.4 Focus and Organisation of the Thesis

1.4.1 Focus of This Thesis

This thesis focuses on the different types of models used to replenish independent demand inventories. These models help the inventory managers answer the questions of how much, and when orders should be placed to meet a stated customer service level at the lowest possible cost. There is no attempt to cover the important topics of MRP and JIT because they go far beyond inventory control and encompass the entire production system. This work is only concerned with order point systems which deal with independent demand inventories.

As mentioned before, the independent demand inventory is subject to market forces. The sophisticated situations in practice have led to hundreds of inventory models being developed. It is practically impossible to include all the models in the knowledge base of a computer-based intelligent inventory system to help inventory managers in making day-to-day decisions. Moreover, the ideal situation, from the viewpoint of building expert systems (ES) or intelligent decision support systems (IDSS), is to have an application in which knowledge is restricted to a very narrow domain and decisions are entirely determined by measurable factors [13] because the more firm our understanding of the expert domain is, the more reliable the system's decision will be. Therefore, to achieve a practical scope, this thesis is limited to a single item kept in one location. In particular, the coverage will not be extended to include the topics of multi-item and multi-echelon systems because much work remains to bring the theory of such systems closer to practical application [14, 15].

For example, in the case of a multiple-item inventory problem, computations of demand and cost data resulting from different items may suggest different inventory models. Under such conditions, if a particular group of items is to be ordered from a particular supplier, it may be the case that all the items in the group do not have the same supply time, then some of the items may have to be ordered earlier or later than their initial optimum replenishment time. Such adjustment in time supply may, depending upon shared cost, lead to a loss of optimisation in relation to their ideal models.

Although considerable work has been done on the multi-echelon inventory problem relevant within a military context, these multi-echelon models are too complicated for routine use by companies with an order of magnitude smaller than the military establishments for which the work was done [11].

These two cases are topics of current research interest and can be embedded on an inventory IDSS, once the research in these topics as well as the experience with developing inventory IDSS start to mature.

1.4.2 The Organisation of the Thesis

This thesis is divided into eight chapters , followed by several appendices.

Following the introduction, Chapter 2 is concerned with a review of the literature which seeks to put the work in its academic context by describing and discussing the research work carried out in the areas brought together by this research. The chapter is subdivided into sections covering work in the fields of inventory modelling, decision support systems (DSS), expert systems (ES), intelligent decision support systems (the integration of DSS and ES), and the development of intelligent systems for supporting inventory management decision making.

Chapter 3 is concerned with the design methodology behind the system developed. It opens with a discussion of activities included in building an intelligent inventory management system. Consideration is then given to the specification of the proposed system to address the problem, and the resulting design. The chapter concludes with comparing the proposed system with other published intelligent (or knowledge-based) inventory systems.

Chapter 4 deals with the identification of inventory demand patterns. First, the classification of inventory demand patterns and the statistical forecasting methods are discussed. Then the process of identifying demand patterns is described.

Chapter 5 is devoted to the inventory models embedded in the system developed during this research. Some models have been developed to cover gaps in the literature of some simple cases. The existing models which address the same inventory problem have been compared and some necessary improvements on the existing models have also been made. The objective here is to adapt the theoretical models to the practical situations.

Chapter 6 covers the actual development of the proposed intelligent inventory management system and the selection of both the hardware and the software. Designs of the system components are also described.

Chapter 7 provides examples of some individual results produced by the developed system when applied to real data supplied by the co-operating companies.

The final chapter considers the research as a whole, discussing achievements and shortcomings, and suggesting some avenues for further work.

This thesis concludes with a number of appendices containing the procedures for developing some inventory models and a user guide for the software.

Chapter 2

LITERATURE SURVEY

This chapter briefly reviews the relevant literature. This review chapter is in six main parts. The first three sections survey the principal domains: inventory modelling, decision support systems(DSS), and expert systems (ES). This is followed by a discussion of the intelligent decision support system (IDSS) which integrates the two fields: DSS and ES. Section 5 reviews the intelligent (or knowledge-based) inventory systems developed in recent years. The chapter concludes with a critique of the published intelligent inventory management systems.

2.1 Inventory Modelling

2.1.1 Brief History of Inventory Modelling

The first mathematical inventory model is generally referred to as the Economic Order Quantity (EOQ) model which was developed by Harris [16] in 1915. The first full length book attempts to explain how various extensions of EOQ can be used in practice is Raymond's [17]. Further research works [18-20] showed that the EOQ model appears to be quite insensitive to errors in the specification of the appropriate cost parameters and the estimation of demand. The importance of the EOQ model is not only from the historical point of view but also because many other models designed to cope with different situations have been based on this model.

However, this mathematical modelling technique of inventory management had very little application at that time. Perhaps this was because the new conceptions always need a period of maturation during which details can be improved upon and the original claim about increased productivity and performance can be proven through the test of time.

The first probabilistic model, now known as the "newsboy problem" or "Christmas tree model" was developed during World War II. Whitin's book [21], published in 1953, was the first text to deal with stochastic inventory models. The first study which analyses a multi-period stochastic model was carried out by Bellman, Glicksberg, and Gross [22].

The fundamental cornerstones of modern inventory theory are the papers by the economists Arrow, Harris, and Marschak [23], and the mathematicians Dvoretzky, Kiefer, and Wolfowitz [24]. In 1958, A full length book devoted to the mathematical properties of inventory models was published by Arrow, Kerlin, and Scarf [25]. From this point on, inventory theory became a full-blown academic discipline, and many hundreds of papers representing developments and improvement have been published.

The appearance of Bellman's [26] "Dynamic Programming" technique helped some real dynamic problems to be solved using this methods. The earliest model based on dynamic programming formulation to deal with non-stationary demand was developed by Wagner and Whitin [27]. Extensions of the Wagner and Whitin model were developed by Zangwill [28].

From about 1963 on, a main research theme has been the concern with multi-item and multi-echelon models. The initial work on the multi-echelon inventory problem is due to Clark and Scarf [29] who assume that *N* installations are arranged in series: each installation supplies the next one in sequence only, and external demand occurs only at the final installation. Significant extensions of this basic model were considered by Bessler and Veinott [30] to deal with the case that each echelon can supply one or more echelons below it.

With the greater diffusion of computers in the 1970s, research has been based more on simulations which use mathematical models to represent inventory systems [31]. The major advantage of this method is that the actual time span over which the real system operates can be compressed and its long-term performance analysed in a few minutes on a computer [32]. By the 1980s inventory DSSs were developed (see Section 2.1.3) followed by attempts to build intelligent inventory systems (see Section 2.5)

2.1.2 Discussion of the Survey Papers and the Modelling Classifications

For over half a century the literature has included frequent publications or research papers on inventory modelling. This proliferation of inventory models has promoted some researchers to publish surveys of the current status of these models in particular and the inventory theory in general.

An earlier survey that mathematically described the status of inventory modelling was published by Veinott [33]. First, the author reviewed the structures of optimal solutions under a variety of assumptions about the cost function with certain and uncertain demands. The different solutions, which depend upon whether the cost function is convex, concave or proportional, were discussed. Then, particular attention was focused on multi-item and multi-echelon problems because the inventory research was dominated by these problems at that time.

Another survey presented by Aggarwal [15] in 1974 also fully described the mathematical models published from 1966 to 1973. The paper first listed the eleven quantifiable variables that affected inventory formulas. Based on similarities of approaches used by the researchers, various papers reviewed were grouped in six categories:

1. Models for deterministic optimum inventory policies,
2. Lot-size optimisation,
3. Optimisation of various specific management objectives,
4. Models for optimising highly specialised inventory situations,
5. Applications of advanced mathematical theories,
6. Models bridging the gap between theory and practice.

Under these categories, a synthesised description of 72 models, covering the range from simple deterministic models to complicated multi-echelon and multi-item models, were presented, and the salient features of individual studies were highlighted. In an attempt to bridge the gap between theory and practice, two simulation methods were discussed in the sixth category. One is Packer's [34] model to find the parameter of exponential smoothing the demand forecast. Another model is developed by Eilon and Elmaleh [35] for computing control limits (s, S values) by simulating the inventory system.

In 1981, Silver [11] published a survey paper entitled "Operations Research in Inventory Management: Review and Critique." Much of the thinking expressed in this paper is condensed from a text written by the author along with Peterson [36]. Following a general discussion of various objectives and constraints faced by decision makers in inventory management, a new category of inventory cost, called system control cost, was introduced. Although the intention of this paper is not to present details of theoretical developments of inventory models, a brief review of standard problems were presented and a number of research problems, including erratic and intermittent demand, service parts, and uncertainty of future demand, were highlighted. The practical solutions presented to these problems would have a major beneficial impact on the practice of inventory management.

The significant contribution of Silver's paper [11] contains the following suggestions given for bridging the void between theory and practice of inventory management:

1. More attention should be devoted by analysts to formulating an accurate model and obtaining a good solution to it rather than getting the optimal solution to mathematically interesting problems. An understandable decision rule that improves somewhat on current conditions is almost certainly better than the optimal solution that is neither understood nor accepted by management.

2. There is a need for easily understood procedures, particularly in smaller organisations. This suggests the increased use of implementation aids such as graphs and tables.

3. A more explicit account should be taken of system control costs in deciding on the form of the control system to be used. In particular data requirements must be realistic.

4. More emphasis should be placed on convincing the decision maker that the decision system is aiding, not replacing, them and that it is in their interest to co-operate.

A relatively recent survey paper of independent demand inventory modelling was published by Tinarelli [37] which discussed in detail the classical EOQ model and its extensions. The author introduced the main costs of inventory and the basic assumptions of EOQ models. They also described the development of several generations of the EOQ model by modifying its basic assumptions. These models, which form the basis of deterministic inventory problems, can deal with real management situations including: EOQ with finite production or supply rate, EOQ with discounts, EOQ with integer quantity, EOQ with allowed shortage, EOQ with constraints, and EOQ with variable holding costs.

The main contribution of Tinarelli's survey [37] to the theory of inventory management is that the author referred to the publications in a systematic manner by classifying them into six groups:

1. Stochastic models,
2. Dynamic demand models,
3. Models for perishables,
4. Joint-ordering systems,
5. Capital and/or volume constraints,
6. Inventory control and devaluation.

Instead of giving detailed descriptions of the contents of each publication, the author discussed the criteria used in compiling this classification and listed the corresponding articles to help both the operational researchers who seek to improve their own performance and the academics who want to be acquainted with the state of the art in this field.

From the above discussion, It is clear that the early published surveys focused on the mathematical descriptions of the developed models, and the more recent surveys aimed at revealing the problems in inventory modelling and seeking methods to promote applications for existing models. The authors reviewed the inventory models from a variety of viewpoints but considering the difficulties to place the publications in one systematic format, all these review articles have helped the research community to gain an excellent overview of inventory modelling.

To promote the application of the published inventory models, some researchers have started classifying the inventory systems to help inventory managers to find an appropriate model from the extensive inventory literature for a given condition.

Similar to the structure proposed by Kendall [38] for classifying queuing models, and based on Naddor's [39] book, Hollier and Vrat [40] presented a classification of inventory systems based on four aspects, namely, structure of the system, environmental parameters, operating policies and inventory-related costs.

The system structure refers to the number of items, the number of suppliers, and the number of echelons. The environmental parameters includes the nature of demand variation, lead time distribution, shelf life and size of back-logging permitted. The inventory policies taken into account are order size, reorder point, review period and order level. The four major inventory-related costs are carrying, shortage, replenishment and procurement costs. This approach can help in classifying any inventory system in terms of all relevant information in a compact form.

Another research effort to classify inventory models were conducted by Barancsi et al. [41]. Based on the study of a sample of 336 inventory models, the authors composed a code system to describe the characteristics of the models. The main codes, each having two or more possible values, are the number of items, the number of locations, input/output character, dynamics, objective, ordering rules, review periods, handling of shortage, and lead time distribution. These codes represent the main properties of inventory models, and their values give more detailed information about the characteristics specified by the main codes. The proposed classification is illustrated in Fig. 2.1, which divides the published models into 12 main classes of models and their subgroups.

The main theoretical result of these studies on the classification of inventory models provide a common terminology to facilitate the understanding of inventory models. These classifications would simplify the selection of an appropriate model for a given set of inventory conditions and lead to possible research ideas, although they have probably not helped practitioners very much.

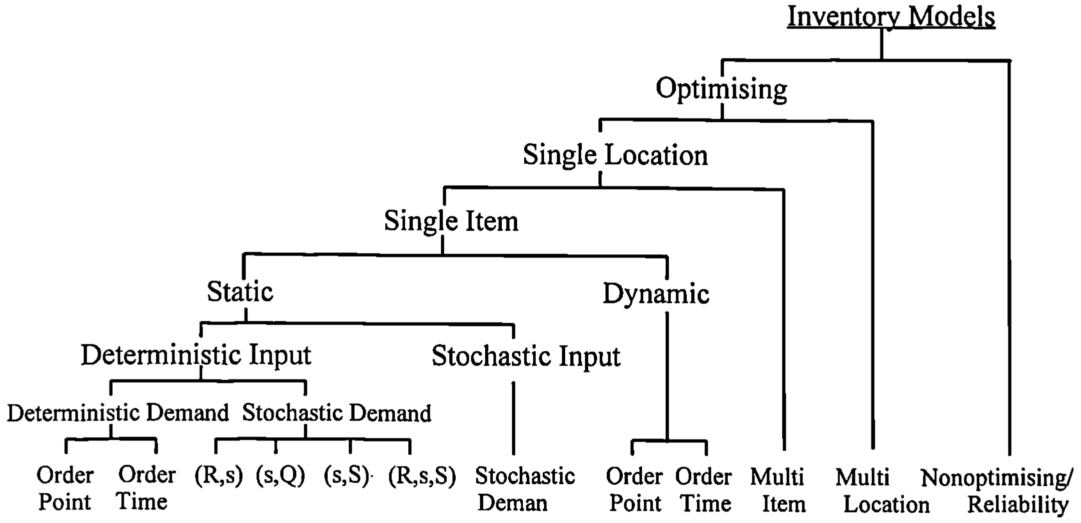


Figure 2.1 A Classification of Inventory Models (Proposed by Barancsi et al [41])

2.1.3 Computerised Inventory Systems

The surveys and classifications of inventory models can provide useful information for decision makers who want to use the models. But the applications of these models in making decisions for inventories with large numbers of items would involve a substantial number of numerical calculations. To facilitate the models application, efforts have been made over the past four decades to develop computerised inventory systems to help in solving these problems.

In 1960s IBM introduced an inventory package called IMPACT: Inventory Management Programme and Control Techniques [42] for the whole and retail trade. Other computer manufacturers followed with very similar systems. ICL [43] introduced its system SCAN; Semens developed its system HOREST [44]. Actually, inventory packages like IMPACT are criticised for not being based on truly optimal algorithms but on heuristic, i.e. common sense accompanied by mathematical solutions for simplified models which involves so many exceptions and restrictions that it is quite difficult to gain insight into the fundamentals [45].

During the 1980s, several inventory decision support systems were developed. In 1985, Kiran and Loewenthal [46] developed a system called "An Integrated Decision System for Inventory Management (IDSIM)" which is a microcomputer based decision support system. It can be used to make decisions on inventory control parameters such as order quantities, safety stocks and order points, at both the individual item and aggregate levels. IDSIM consists of a main program which interacts with six other functional subroutines that are transparent to the user. The most recent version of IDSIM, which is available for the APPLE II$^+$ and APPLE IIe, is written in an interactive mode in order to facilitate its utilisation in the decision making process. This allows the user to alert the data and constraints and to perform sensitively analyses at various points during the program execution.

Another system called MICROLOT: Inventory Control System On The Microcomputer, developed by Moras [47] in 1989, implements a decision support system for single-item inventory problems with deterministic and stochastic demand distributions. The deterministic case includes systems with constant (linear) demand and systems with time-varying demand. Periodic review (order-up-to-level) and continuous review (order-point) systems are addressed in the stochastic case. It also contains a Monte Carlo routine which simulates stochastic systems and can be used to verify theoretical results. The program is written in BASIC, and runs on IBM and compatible microcomputers. It features a friendly user interface and graphical aids.

In order to deal with service part inventory problems, Hill et al. [48] successfully implemented a DSS for analysing service parts inventory retention stocks. The system was installed in a Fortune 100 company in the United States with an initial $50,000,000 investment in service parts inventory. This system contributes to the service parts management in three ways. First, it proposes a new forecasting model for the retention stock problem. second, it develops a new inventory model which capture a richer operating environment. Third, it integrates inventory models into an interactive, menu-driven, database DSS. Although the system is developed for a specific company, the authors suggest that it is applicable to managing service parts in a wide variety of environments.

All the systems mentioned above are used to help managers to calculate order quantities. They all seem to lack the ability to select suitable inventory models automatically by analysing historical data and identifying practical situations and to provide the explanation for the model selection process. In recent years, with the emergence of expert systems, some intelligent (or knowledge-based) inventory systems have been developed to provide the inventory managers advice on model selection after asking them a number of questions pertinent to the problem situation. Intelligent inventory management systems will be reviewed in Section 2.5 following a short introduction and review of DSS and ES in general in the following sections.

2.2 Decision Support Systems

One of the characteristics of DSS is that they are oriented towards evolution and change [49]. The DSS field is part of an evolutionary process in the development of information technology (IT) and management science/operational research (MS/OR) support for management decision making.

During the late 1960's and early 70's technological advances such as the minicomputer allowed the decentralisation of computing resources [50] and permitted the application of interactive computing technology to managerial tasks in order to use computers to aid better decision making [51]. This focus on practical decision support was made possible by the synthesis of the work from two research areas: theoretical studies of organisational decision making and interactive computing technology.

The first explicit synthesis of these two contributions was Management Decision Systems by M. S. Morton [52] which described an experimental project investigating the use of the new interactive technology in supporting management decision making. This injected a behavioural perspective into computer science and introduced the technical innovations of interactive computing into organisation structures.

During the first half of the 1970's the main work on the conceptual foundations of decision support was carried out and some experimental systems were built by academics in collaboration with innovative managers [53]. Various names were used for these computer-based systems. The term of DSS was first used in the paper by Gorry and Morton [54] which combined classifications of managerial activity within organisations as developed by Anthony [55] and Simon [56] to provide a framework for computer-based information system in organisations.

Keen [57] describes DSS research gaining momentum around 1974 and having reached a "critical mass" and expending beyond a fairly narrow circle by 1978. In this year the first influential DSS books, *Management Decision Support Systems* [58] and *Decision Support Systems: An Organisational Perspective* [50], appeared. The other key early text was Building Effective DSS [59] which focused more on practical implementation frameworks than the earlier texts.

A survey of the literature in the DSS field reveals a wide range of definitions of what a DSS should consist of, what it should do, and how it should attempt to do it. The synthesis of definitions of DSS provided by Emo and Lee [60] describes DSS as a computer-based interactive system that

- supports decision makers rather than replace them
- utilities data and models
- solves problems with varying degrees (structured and ill-structured)
- focuses on the effectiveness rather than the efficiency of decision processes.

Just as there is no widely accepted definition of DSS, there are many proposed frameworks for research into and development of DSS. In an influential framework, Sprague and Carlson [59] identify the key components of DSS as:

- Data base (external data, internal data, and data base management system)
- Model base (decision models, model base management system)
- Dialogue system (interface).

Instead of the recognisable evolution of MS/OR into DSS, the current AI/ES technology (discussed in next section) is also being incorporated into DSS [61]. Turban and Watkins [62] quote S. Morton, one of the DSS pioneers, as suggesting in 1984 that " DSS as we know them may even become obsolete in the foreseeable future. They are being supplanted by expert decision support system. The next generation of DSS will combine existing DSS technology with the capabilities of AI." Alternatively, Wynne [63] suggests a sequential progression from MS/OR through DSS to ES with each stage dominating parts of the previous one. These early claims and hopes have encouraged intellectuals to carve out of new fields.

2.3 Expert Systems

2.3.1 Definition

A formal definition of ES approved by the British Computer Society's Committee of the Specialist Group on Expert System [64] is as follows:

An expert system is regarded as the embodiment within a computer of a knowledge-based component from an expert skill in such a form that the system can offer intelligent advice or take an intelligent decision about a processing function. A desirable additional characteristic, which many would consider fundamental, is the capacity of the system, on demand, to justify its own line of reasoning in a manner directly intelligible to the enquire.

Such a program should give "a high level performance on problems that are difficult enough to require significant human expertise for their solution" [65]. In particular, Expert systems are required to solve complex problems in a specific knowledge area and show the logical steps leading to the solution.

2.3.2 Structure of Expert Systems

Basically an expert system has three major components (Fig. 2.2): a knowledge base (KB), an inference engine, and a user interface. The knowledge base of a classical expert system is a collection of representations of knowledge extracted from experts in the domain. The inference engine uses the knowledge base to solve the problem. The user interface enables interactive communication between the user and the expert system.

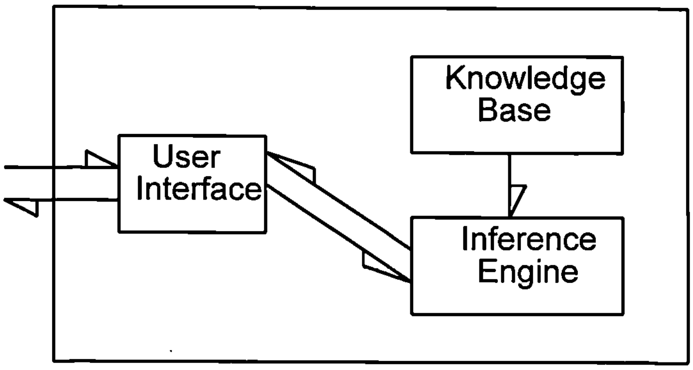


Figure 2.2 Expert System Structure

The KB is at the heart of an expert system. The power of an expert system is derived primarily from the wealth of specific knowledge of a domain that can efficiently be encoded and not from the strength or ingenuity of its reasoning methods. Thus, " expert systems must be knowledge-rich even if they are methods-poor" [66]. For this reason, expert systems and related program are called knowledge-based systems (KBS).

A crucial factor in the design of a KBS for a particular domain is knowledge representation, i.e. how the knowledge is encoded within the program. The most commonly utilised representational techniques are the semantic networks, frames, and production rules [67].

The semantic network is a classic representation technique used for prepositional information. It was first developed for representing human memory and language understanding. The structure of a semantic network is shown graphically in terms of nodes and the arcs connecting them. Nodes are circles and arcs are the lines connecting nodes. Nodes are often referred to as objects and the arcs as links used to express relationships that provide the basic structure for organising knowledge. Two kinds of commonly used links are IS-A and A-KIND-OF. The link IS-A means "is an instance of" and referred to a specific member of a class. The link A-KIND-OF is used to relate one class to another.

A frame is a structure containing information about a single entity. It consists primarily of a set of slots and fillers. A slot is a component of a frame that refers to a specific attribute of the frame entity. The fillers may be values or a range of values of slots. A frame system would be a good choice for describing a mechanical device such as an automobile. A frame is analogous to a record structure in high-level languages such as Pascal and LISP. Corresponding to the fields and values of a record are the slots and slot fillers of a frame.

The production rules or if-then rules are characterised by two clauses: the antecedent or if clause, and the consequent or then clause. The production rule formalism is a flexible mechanism that can be used to represent several types of knowledge such as situation/action, premise/conclusion, and sufficiency. In most successful expert systems knowledge is often in the form of production rules because they can offer outstanding advantages including [68]:

1. Modularity: each rule defines a piece of knowledge in a way which is significantly independent from other rules.

2. Flexibility: it is quite easy to add new rules to the knowledge base without affecting the rules already present.

3. Modifiability: if some rules are to be changed, they are easily modified with little influence on the other ones.

In addition to the rules, the knowledge base contains factual knowledge, in the form of facts, that represents the information generated through the dialogue with the user or other assertions of a general nature.

The inference engine contains the control strategy for drawing a sequence of inferences from the knowledge base. Two well known control strategies [69] are backward chaining (goal driven) and forward chaining (value driven).

Backward chaining starts with a goal expression whose value must be determined, the engine searches the KB for rules that allow it to conclude a value for the goal. These then serve as the new goals and the search continues. In the process, the system is using rules and facts of the KB and may ask the values of certain expressions from the user. Backward chaining is found in a number of expert systems since it generates a more directed sequence of questions. Forward chaining starts with the known values of certain expressions and uses these to infer the values of other expressions until a goal is reached. Forward chaining is used in expert systems that accept information directly from measuring instruments or users. It has been recognised that the designer can use rules to influence the control strategy of an expert system. These guidelines (meta-rules) may direct the search to specific portions of the KB. Some of the researches on meta-rules is reported in Buchanan and Shortliffe [70].

The user interface consists of tools to help the user to interact with the system. The design of a user-friendly interface assumes even greater important in microcomputer-based systems [71]. This is a major factor in the acceptance of an expert system. Therefore, some researchers [72-74] have extensively discussed the interaction between the user and the program. One important component of this interaction is the questioning strategy. Most users would feel uncomfortable with an apparently random or aimless string of questions and quickly lose interest in the program. Another form of interaction with the user is explanation. In rule-based systems such explanation implies that the reasoning process should be available to the user for easy inspection.

2.3.3 Tools for Developing ESs

In theory, expert systems can be developed by traditional languages such as PASCAL, Modula-2 or C, which have the advantages of being faster, more compact and portable and of being the languages that are more common than their AI counterparts within commercial data processing environments. However, these traditional tools are poorly adapted to the expression and the handling of knowledge expressed by symbols. Some specific languages that are developed specially for AI, like LISP and PROLOG, feature desirable AI facilities which include list processing, recursion, frame-like data structures, the ability for programs to act as data, production rule formalism and logic programming. However, these languages are difficult to interface with most of the existing commercial systems which are widely written by traditional languages. In addition to these, with a programming language the knowledge engineer must start from the first principles and construct his own knowledge representation formalism, inference mechanism and user interface routines. This approach is the most flexible but is also the most time consuming.

Another approach for developing ESs is to use expert system building tools (or shells), which are software associated utility programs to facilitate the development, debugging, and delivery of application programs for certain types of applications in which the user must only supply the knowledge base [67]. These shells have advantages in knowledge representation and inferring, but have weak calculation and designing graphical user interface abilities.

In the past decade a number of ES shells have been developed from essential abstractions over one or more application programs. The significant features of some significant and recently released ES tools are briefly discussed as follows [75-77].

The Kappa 3.0 is one of the developing tools for Unix to support distributed applications which range from the provision of general-purpose computing facilities for group users to automated banking and multimedia communication systems, and embrace all the commercial and technique applications of computers. This tool includes a graphical user interface (GUI) builder. It also provides full C^{++} interoperability. Kappa 3.0 allows different developers to work on segments at an application and then merge the modules. Combining with the Kappa ComManager, an entire application can be developed on one platform and application modules can be split into GUI, logic, data, or in between to run on different platforms.

ART enterprise is the latest incarnation of ATR-IM. ART enterprise provides portability across a range of platforms, including Windows, OS/2, Macintosh and the major Unix workstations. Its GUI based development environment will adopt the look and feel of the native windowing environment. Automatic integration of multiple relational database management systems means support for query, join and update of distributed data across multiple databases. Other features includes automatic application generating tool to assist rapid prototyping.

HipWork is an ES tool which enables developers to build large-scale and intelligent multimedia applications. It is among the first hypermedia environments for Unix workstations. It integrates rule induction tools, neural networks, object-oriented technique, and database. HipWork allows developers to mix text, sound and graphics for user interface, and offer libraries and collections of palettes and icons to help in the construction of on-line documentation and demonstrations.

Leonardo for Windows is a major new version of the expert system development environment. At the core of the new release (Leonardo 5.0) is a proprietary object management system which combines the best features of Smalltalk and C^{++}. Its integral procedural programming language has been enhanced with new functionality. It has new data types, new functions for handling objects and attributes, and improved integration between the procedure language, inference engine and screen designer. The new screen designer includes new tools for building end-user help screen.

Casework is a new case-based reasoning tool running on PCs under Microsoft Windows. It can be used in conjunctions with the rule-induction shell, KATE. Casework automatically maintains the applications while new cases are added. it can optimise its reasoning when there are unknown or missing values in a consultation case, and it implements a fuzzy information retrieval system. CBR2 is the recent generation of a family of case-based reasoning (CBR) tools specialised for customer support and help desk automation. It allows the integration of documents as well as the cases into a knowledge base under a single graphical user interface.

Each expert system tool has its own significant features and is specialised for developing particular applications. Choosing the right developing tool is a vital step in prototyping an expert system. Table 2.1 highlights the characteristics of the tools mentioned above. However, ES tools are continuously developed and enhanced at a high rate. Generally, there is a substantial enhancement every year in each tool and a major revision every two to three years. In addition, a new generation of software tools is slowly emerging from the research laboratories which attempt to help in progress of expert systems construction without cramping the knowledge engineer's style [78]. The components of these tool kits are various architectural building blocks, task-oriented representation language, and some kinds of system philosophy that helps the user put them all together.

Table 2.1 Summary of the Characteristics of Some Developing Tools

Tool **Attribute**	**Kappa 3.0**	**ART enterprise**	**HipWork**	**Leonardo**	**Casework CBR1**
Hardware platform	Work station	PC, Work station	Work station	PC	PC
Operating environment	Unix	Unix, OS/2, Windows	Unix	Windows	Windows
Knowledge base	Rule based	Rule based	Rule based	Rule based	Case based
User interface	GUI	GUI	Hypermedia	No GUI	GUI
Integration	C^{++}	Database	None	Smalltalk, C^{++}	C^{++}
Other	support distributed applications	support exchange data across databases	On-line documentation and demonstration	tool for building end-user help screen.	maintain case base

2.3.4 ES Applications

Expert (or knowledge-based) systems are one of the success stories of artificial intelligence research. Although the enthusiasm of theoretical research is cooling down, practical applications are attracting increasing attention. In a recent survey conducted by the UK Department of Trade & Industry reveals over 2000 ESs in commercial operation [79]. Probably the three most important expert systems from a historical respective are MYCIN, PROSPECTOR, XCON. Each had a significant impact on the field.

MYCIN [80] is the most famous expert system for diagnosing microbial diseases of blood. It employed a knowledge base of IF-THEN rules and is the classic backward chaining system. The confidence measurements are introduced to help in handling the imprecision involved in diagnosis. The confidence manipulations are based upon fuzzy logic.

PROSPECTOR [81] is an expert system for geological exploration. An important contribution of the PROSPECTOR system is the use of inference networks. An inference network is a tree where the nodes correspond to assertions. PROSPECTOR has nodes that compute the probability values from their inputs in three ways: AND nodes, OR nodes, and weighted combination nodes. The AND and OR are calculated using fuzzy logic approach of substituting MAX and MIN respectively. The weighted combination nodes involves more complex calculation in line with the Bayesian probability theory.

XCON [82] is the first major success story of commercial expert systems. XCON configures computers, a difficult and complex task that used to be performed by proficient computer engineers. XCON is a classic forward chaining system programmed in OPS5. The program begins with a set of values for initially known variables and then repetitively performs recognise-act-cycles. Sometimes the action of a rule is to prompt the user for additional information. The system had been in daily use by Digital Equipment Corporation (DEC). A team of programmers maintain it and continue to enhance it as new hardware becomes available.

The successful stories of applying the new technology in the medical and industrial fields have promoted some researchers [83, 84] to review AI/ES concepts for the MS/OR community and provide some possible research topics for using expert systems in management science such as model choice, search for solution, interpretation of data, etc. This has led to a number of expert systems [85-91] being already developed in the fields of accounting, finance, quality control, forecasting, statistics, maintenance and inventory.

2.3.5 Advantages and Limitations of ESs

Expert systems can be used to solve problems that are extremely difficult to solve by conventional programming techniques [80]. To compare with human experts and conventional software, expert systems have some significant advantages [92, 93]:

1. The kind of problems an expert system solves can not be solved by a person who has not been trained in the domain of the problem.

2. Expert system methodology separates the expertise itself (knowledge base) from the mechanism which applies the expertise (inference engine).

3. Expert systems can preserve valuable knowledge which would otherwise be lost when an expert is no longer available.

4. Expert systems can enforce consistency.

Despite the undoubted success of KBS in many sectors, these systems should not be thought of as a general panacea. No technology can solve all problems, and therefore expert systems still have some limitations [94, 95]:

1. The limited learning capacity; the current commercial expert systems do not include significant learning capabilities.

2. The difficulty of presenting common-sense knowledge.

3. It is unlikely that an expert system can ever contain the complete expertise and experience of a leading expert, an expert system can only be developed for a narrow and

well-bounded domain that does not require much common sense or encyclopaedic knowledge.

2.3.6 Recent and Expected Future Developments in ESs

The term of "second generation expert systems" is collectively used to refer to the recent developments in KBS. The second generation expert systems are characterised by two approaches: combining multiple models and reasoning techniques [96], and using knowledge-level approach [97] for designing systems. Explicitly modelling different types of knowledge is a defining characteristic of the second generation expert systems in utilising a variety of problem solving methods geared towards particular subtasks. The motivation for designing KBSs at the knowledge level is not only to provide a more accurate description, but also to emphasise how the problem should be solved rather than how the system will be implemented. In particular, second generation expert systems demonstrate the importance of using the appropriate knowledge for given problems, and representing that knowledge in appropriate ways. Such systems often include multiple representations, problem solving strategies, and learning methods within a single system. However, there is no clear distinction between "first" and "second" generation systems: it is more of an evolution of ideas, styles, and techniques for constructing KBSs [98].

Over the last few years an alternative reasoning paradigm and computational problem solving method, called Case-Based Reasoning (CBR) [99], has increasingly attracted more attention. CBR solves new problems by adapting previously successful solutions to similar problems. It could be applied to solve a problem where no explicit model existed, and can learn by acquiring new cases and so improving its performance with time. In the UK, CBR seems to be particularly applied to civil engineering. A group at the University of Salford are applying CBR techniques to fault diagnosis, and the repair and refurbishment of buildings [100]. Yang & Robertson [101] in Edinburgh are developing a CBR system for interpreting building regulations.

Recently, an interest has emerged to combine KBS with Neural Networks [102] to generate a neural network expert system. This hybrid system will help developers to generate knowledge base automatically from training examples or machine learning. Nikolopoulos and Fellrath [103] have developed a hybrid system in the financial domain to corporate neural network and expert systems in a unified model to produce investment strategy recommendations. In their work, the neural network is used to predicate necessary economic environment information for the expert system's knowledge base.

Generating and debugging the knowledge base is the most difficult and time-consuming task in KBS building process. The future development may introduce an approach to automate the creation of knowledge base [104]. The method will produce a set of rules with a number of instances of questions with answers which would be supplied by an expert. But there is an inherent difficulty in that the rules generated by this method may make no sense at all to a human expert.

2.4 Intelligent Decision Support Systems (IDSS)

ESs contain large knowledge bases of domain knowledge gathered from domain experts and "support the work of professionals engaged in design, diagnosis, or evaluation of situations requiring expertise in a particular domain" [105]. Knowledge collection, usually requiring the help of a knowledge engineer, is very time consuming and expensive. The data (or knowledge) is historical in that it represents how experts have reacted in situation encountered or been trained to react in situations envisaged. These factors limit the usefulness of ES for dealing with the type of novel, one-off questions encountered in low-structured decision making environment that DSS set out to address. Therefore, it can be argued that ES as "standalone" consultation aids are not of interest from a decision support point of view since they make use of existing information rather than supporting the search for new. The integration of ES with MS/OR models and DSS is however of considerable interest and potentially of great value [62].

Eom and Lee [60] in their survey of DSS applications notes the incorporation of many AI/ES techniques as important DSS tools. Phelps [106] commented that the integration of ES and DSS combines "the data exploration techniques and decision theoretic techniques of statistics, the problem formulation and optimisation techniques of OR, and the ES approach of AI to make a new and more powerful form of decision aid." Such integration systems may be termed intelligent decision support systems. The integration of DSS and ES is also seen as an important research theme by Watson [105].

As well as injecting intelligence into each DSS component, O'Keefe [107]concludes that the MS/OR community should regard ES techniques as another modelling tool, "allowing the construction of knowledge-based symbolic models rather than the traditional mathematical or statistical models." Dewhurst and Gwinnett [108]proposed the construction of a "decision-analysis support system" with an ES shell, along with OR techniques, statistical analysis and a simulation language, as one of the modelling techniques available to the user. The integration of ES with DSS is greatly expending their capabilities. Specifically, combined ES-DSSs are applicable to a wide range of applications, can handle more steps in the decision making process, and can fit more diversified and complex decision situations [61]

The nature of the IDSS and the method of integration of AI/ES techniques depend on the requirements of the area of application [62, 109]. AI/ES techniques potentially of use in the development of inventory IDS system include incremental development, knowledge representation, knowledge base management, intelligent user interfaces and explanation (logic-tracing). The benefits of the integration include the development of more active, participative and valuable DSS which can be used to answer "Why" and "What-if" questions.

Within the area of information systems DSS is a mature field and promising new developments are arising from the integration of more recent techniques such as knowledge-based systems. Such integrated systems may provide a means to start to address range of problems including the automatic selection of an suitable inventory model based on the mathematical analysis of inventory data arising from the operation of large-scale industrial organisations. The IDSS approach may offer a novel opportunity to tackle this problem through the incorporation of mathematical inventory models in a DSS with some "intelligent" features.

2.5 Intelligent Inventory Management Systems

Inventory management is a complex problem area due to the diversity of real life situations. Successful inventory management requires sophisticated methods to cope with the continuously changing environment. The current applications of expert systems have demonstrated quite successful results in terms of better and more effective decision-making and management. This has promoted some researchers to develop some intelligent (or knowledge-based) inventory management systems for this potentially useful area.

In 1988, Hosseini et al [110] introduced A Knowledge-based System for Inventory Control (EXIS) to recommend optimal inventory control decisions. The knowledge base of this system consists of five models for constant demand and two models for stochastic demand (Fig. 2.3).

The deterministic models that are embedded in this system are able to deal with the simple constant demand inventory problems, but they cannot cope, for example, with complicated situations such as backorder or lost sale cases under finite supply rate condition. Moreover, in the above five deterministic models an important factor, namely, quantity discounts is neglected despite the facts that in many practical situations purchasers can get unit price discounts if the order is larger than certain amount (breakpoint) set up by the supplier. For the two stochastic models, the only difference between them is how to calculate the order quantity. Although the policy selected for stochastic demand was not explicitly stated, we can infer that the (*s*, *Q*) system is adopted from the equations cited in the paper.

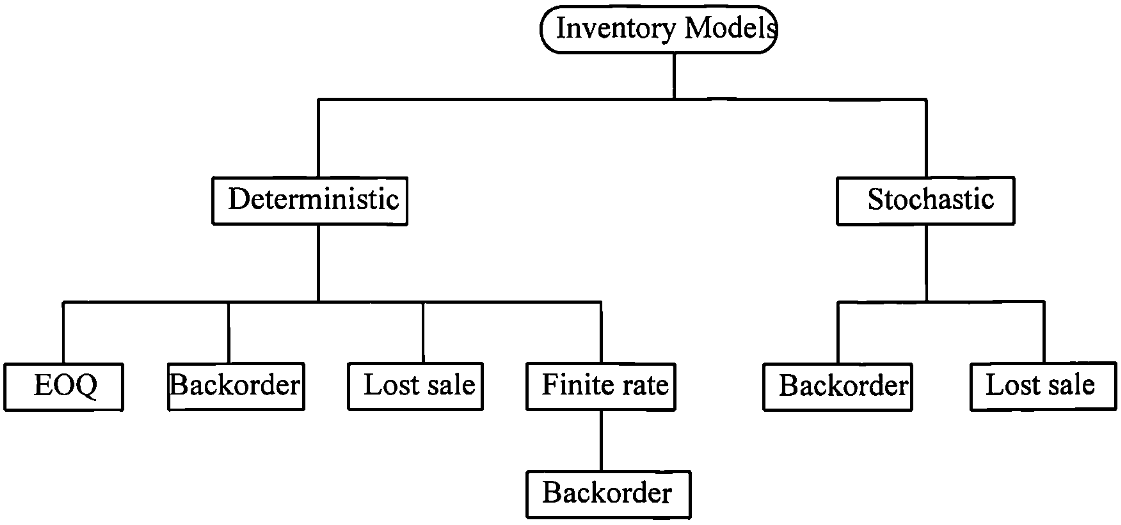
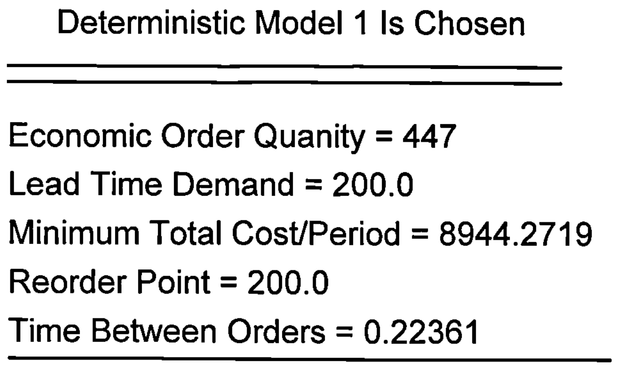


Figure 2.3 Inventory Models Included in Hossseini's et al System [90]

In order to give advice about the optimal order quantity (see Fig. 2.4), the user must give the system reliable information about the relevant parameters used to identify the correct inventory model for calculating the order quantity including the demand type, the supply, the costs, the lead time and so on. The system can not explain why and how the model is selected which is a fundamental characteristic of expert systems.



```
Deterministic Model 1 Is Chosen

Economic Order Quanity = 447
Lead Time Demand = 200.0
Minimum Total Cost/Period = 8944.2719
Reorder Point = 200.0
Time Between Orders = 0.22361
```

Figure 2.4 Sample Output for Model Selection Module (Hossseini's et al System [90])

EXIS was developed in OPS5 and several external functions are written in FRANZLISP, mainly because of the developer's experience with the language. The system runs on a Digital Equipment Corp VAX 11/75 minicomputer with VT100 video terminal. It contains a control module, a model selection module, a calculator module and a user interface for data entry, monitoring and updating inventory position. These modules communicate with the inventory models included in the knowledge base of the system to find an optimal inventory decision.

In 1989, Sinha et al. [111] presented a conceptual design of an inventory expert system which implements a set of rules for the selection of inventory models, and presents the output through tables, plots or graphs. The scope of the system is limited to single item, single location.

The inventory models which intend to be included in the knowledge base can cope with items with deterministic demand and items with probabilistic demand with known probability distribution. It was claimed that three control policies, (*s, Q*) system, (*s, S*) system, and (*T, s*) system, will be adopted by the system. But these policies have different properties depending on the specific environment in which they are implemented. To incorporate them into one system can complicate the system and confuse the users because the system itself does not have the ability to suggest which policy to be selected according to the practical situation. In addition, the authors also conveyed some incorrect ideas.

First, the authors confused the two kinds of parameters required to select the mathematical models, i.e. the parameters for model selection and the parameters needed for the order quantity calculation. e.g. the variables of review interval, order point, unit price, ordering cost, and holding cost were put on the list of parameters required in selecting an inventory models.

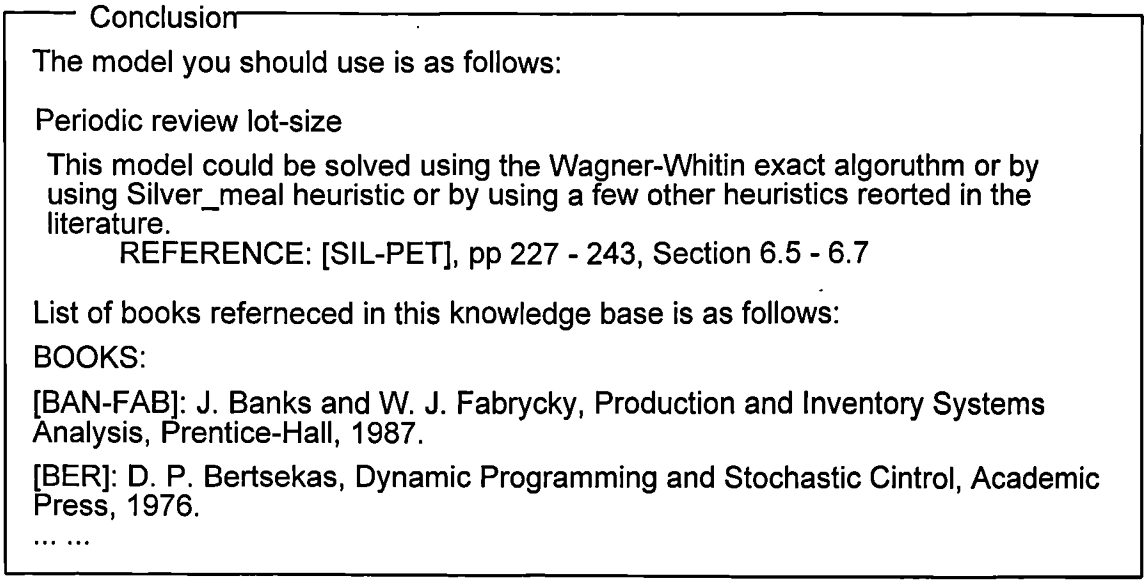
Secondly, in discussing the essential features of an expert system for inventory, the authors pointed out that " the system must be able to construct new models from more primitive models provided in the model base." we believe that it is impossible to realise this idea at the present stage of expert systems technique.

Finally, there are no details in this paper about the system structure, the implementation process and the kind of tool (or shell) to be used for development. This means that the authors have not experienced any real development of KBS for inventory management. There is a great distance between conceptual imagination and practical development of an inventory expert system.

Parlar [112] discussed the development of a knowledge-based expert system which can identify and recommend up to 30 production-inventory models found in almost every major production-inventory textbook. It is claimed that this system can correctly advise on model selection and supply the inventory manager with a literature reference for a suitable model, based on the user's responses to the system's queries. This system does not have the facility to implement the models. It only serves as a computerised model selection tool.

The system is developed using PC Easy and runs on IBM PC. The PC Easy is a production rule-based shell using a combination of backward-chaining and forward-chaining control strategies. The knowledge base consists of IF-THEN type rules which are chained to each other. The rules in PC Easy are constructed by means of 23 parameters, such as DEMAND-LEVEL, DISCOUNT-TYPE, FIXED-COST, etc. The 68 rules are mainly grouped into four mutually exclusive categories: (1) EOQ related group, (2) Deterministic non-EOQ group, (3) Newsboy model related group, and (4) Stochastic non-newsboy group. Given the value of the parameter GENERAL-TYPE provided by the user the system selects one of the four groups, thereby effectively eliminating all the other irrelevant rule groups. This would cut down the search space of the problem and reduce the time required to find the correct model. It is claimed that the system can be easily extended and refined by adding new rules to the knowledge base when new models become available.

The reasoning procedure starts with prompting the user to input the value for the parameters that contain the information that the system uses to arrive at a conclusion. Based on the user's responses, the system will provide the literature reference for the selected model as shown in Fig. 2.5.



Conclusion

The model you should use is as follows:

Periodic review lot-size

This model could be solved using the Wagner-Whitin exact algoruthm or by using Silver_meal heuristic or by using a few other heuristics reorted in the literature.

REFERENCE: [SIL-PET], pp 227 - 243, Section 6.5 - 6.7

List of books referneced in this knowledge base is as follows:

BOOKS:

[BAN-FAB]: J. Banks and W. J. Fabrycky, Production and Inventory Systems Analysis, Prentice-Hall, 1987.

[BER]: D. P. Bertsekas, Dynamic Programming and Stochastic Cintrol, Academic Press, 1976.

... ...

Figure 2.5 The Conclusion Screen of Parlar's Inventory ES [92]

A deficiency of this system is that it is incumbent on the user to search the literature to locate the corresponding inventory control policy. Therefore, this system perhaps cannot offer a great advice on model selection for the inventory managers who are not quite familiar with the knowledge of inventory modelling or their applications.

To overcome the deficiencies of Parlar's system Luxhoj et al. [113] presents a prototype KBS where the knowledge base contains the control policy equations for reorder level, reorder quantity, and total system cost. The system inherits Parlar's original concept with enriched knowledge base and provides more precise information in line with the inventory model recommended for a specific situation which will lead to more accurate usage of inventory models. After identifying the inventory model the system accesses an external computer program to numerically solve the model equations that the rules select.

The system is developed using PC Plus that is both a frame-based and rule-based expert system shell. In this specific application, the knowledge base is compromised of chained IF-THEN rules. It was claimed by the authors that "Although this KBS prototype only considers the classical EOQ model and its extensions, the knowledge base could be expanded to include more diversity."

2.6 A Critique of the Published Intelligent Inventory Management Systems

In conclusion, there is no paradox in saying that some prototype intelligent inventory management systems (or inventory expert systems) were developed during the last decade, but too often they were produced simply as an intellectual exercise.

First, the knowledge bases of these systems were not properly prepared, perhaps due to the fact that the authors do not have the necessary background and knowledge of the inventory domain. There were no inventory models included in Sinha's et al [111] conceptual system and Parlar's [112] system. Sinha et al [111] did not build an actual model base because the paper just presented a conceptual design of an inventory expert system. The knowledge base in Parlar's [112] system is only a collection of inventory bibliography to help users to find out inventory models in papers and books. The inventory models built in the Hosseini's et al [110] system and Luxhoj's et al [113] system were limited to the items with constant or stochastic demand. For the KBSs, the system functions are mainly dependent on the type of knowledge that has been built in the knowledge base. Therefore, an improper knowledge base will result in poor ES.

Secondly, all of these systems seem to lack the ability to automatically elect suitable inventory models by analysing historical data and identifying practical situations. To use these systems the user has to know what the situation is for every item. The systems can only make a decision after the user has provided all the values of parameters for selecting models. This makes the system just like a classification of inventory models by using computers or at the best like commercial inventory management packages. Moreover, if the systems can not drive parameters required in selecting the model, they are also unable to switch to a new model for an item when its demand pattern has changed.

Thirdly, the problem of how to interface with the existing inventory information systems had been neglected despite the fact that 90% of companies [10] use computers for inventory management purpose. If the new KBS can not access the database of the established systems, it will stagnate the implementation of the inventory expert systems simply because the creation of a database for the new KBS will increase the system's investment. It is also a time-consuming task.

Finally, these published systems seem to be academic exercises. To demonstrate that a new method can solve a problem it should be based on a real world situation. There are no examples given by any of the authors to show the correctness and efficiency of their systems.

Expert systems flourished in the 1980s. The early successful stories have promoted new applications with this technique in many areas [113]. The current applications have demonstrated quite successful results in terms of better decision-making [114-116]. As discussed above, to reap the benefits of this new technology some intelligent inventory management systems (or inventory expert systems) have been developed since 1988 to help manage large scale inventories. These efforts have engendered widespread hopes of effective computer solutions to inventory problems. However, there is still a lot of theoretical and practical work needed to be done in order to achieve this goal.

Chapter 3

OVERVIEW OF THE PROPOSED INTELLIGENT INVENTORY SYSTEM

This chapter introduces the outline structure of the proposed intelligent inventory management system. The first section briefly discusses the objectives of the system described in this thesis. The methodology of developing the proposed system is discussed in section 2. This is followed, in the third section, by the description of the system structure. The fourth section is devoted to highlight the salient features of the system in comparison to other published systems.

3.1 The Objectives of the Proposed System

The few intelligent inventory (or expert) systems published in the literature are based on academic research work. As explained in Chapter 2, they were primarily designed to facilitate inventory management to the users who are not familiar with the wide range of inventory models which were developed for different conditions. They are not adequate for practical applications which requires an automatic identification of a suitable model. If a system is designed to ask the user to feed in all the values of parameters in order to select a suitable inventory model, such a system can be easy to develop, but may not have any practical value.

To overcome the drawbacks of the published systems and in order to achieve an efficient intelligent inventory management system one should answer the following fundamental questions. First, does the system require responses from the user to questions in order to select a suitable model? Secondly, how are the parameters used to choose suitable models to be estimated? Thirdly, which inventory models should be included in the knowledge base? Finally, how should the system communicate with the user and access other management information systems? The responses to these questions cannot be satisfied unless a detailed study is made of inventory modelling, quantitative forecasting, existed knowledge-based inventory systems, and the available tools to develop an intelligent inventory IDSS.

The main objective of this research is to develop such a system which has an appropriate knowledge scope (or model base) and focuses on the interrogation of the history data rather than on asking the user to describe the system under analysis. In addition, the tool selected for developing the system must be compatible with the popular software to allow the proposed system to communicate with other information systems. It is hoped that this study can properly answer the above questions and hence develop a system which can offer a new approach to provide a solution to the inventory management problem.

3.2 Methodology

Historically, computers have played a major role in the business world and will continue to expand that role as advances in hardware and software are realised. The evolution of the computer's role began with electronic data processing (EDP) and has progressed through the stages of management information systems (MIS), decision support systems (DSS) and expert systems (ES).

EDP [117] uses predetermined procedures to manipulate huge volumes of data. The available data is manipulated by the EDP so the result can be interpreted by the decision maker to solve the problem. MIS [118] were the next step. The purpose of an MIS system is to summarise and select data from the massive data files so as to give meaning to the computer output. MIS systems are very good at dealing with situations that have a known problem structure. DSS [119] are designed to help managers make more effective and efficient decisions. The DSS is designed to help managers deal with problems where their level of ability to solve the problem is below a certain threshold. ES were evolved from cognitive psychology and information systems [120]. In the area of MS/OR, expert systems are particularly useful in situations involving the choice among several models, the formulation of the selected model and to help provide interpretation of the model result.

In general, an intelligent (or expert) system is designed to use knowledge (or model base) and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions. It is distinguished from other types of computer-based information systems by employing knowledge of the techniques, information, heuristics, models, and problem-solving processes that human experts use to solve such problems. The core phase in developing an Intelligent system consists of three major aspects; knowledge acquisition (to find the expert knowledge used to perform the task of interest), knowledge representation (to represent the elements and the structure of the expert knowledge), and knowledge implementation (to implement the acquired knowledge into a computer program based on knowledge representation developed). The methodology of the research described in this thesis is the practical development of the software called intelligent inventory management system, and the verification with real data to demonstrate and evaluate the system. The integration approach, particularly adapting and incorporating a pattern identifying component and rule base component into a unified system to integrate the data collection, parameter estimation, model selection and order decision functions is the central idea behind the system developed in this project. This makes the system more applicable because it greatly reduces the system reliance on the user responses in recommending optimal decision. Building such an efficient intelligent inventory system management requires a coherent strategy of combining the computer technology with quantitative methods.

3.3 The System Structure

The system described in this thesis aims at constructing an intelligent inventory management system which combines automatic pattern identification and model selection facilities. Such a hybrid intelligent inventory management system consists of a collection of techniques (or pattern identifier) for identifying demand and lead time patterns and a knowledge base (or rule base) for subsequent selection of a suitable inventory model which takes into consideration aspects of the practical situation.

The outline structure of the proposed system is demonstrated in Fig. 3.1. The data manager records and manipulates the historical demand data and other useful information. The pattern identifier then analyses the historical data to identify the demand patterns of inventory items. The output of the pattern identifier is stored as facts for model selection. The monitor is used to check inventory status and generate replenishing reports. The model selector chooses a suitable inventory model based on the user responses and the facts produced by the pattern identifier. Finally, the interpreter reports the order decision computed by the calculator. The result of such harmonious co-ordination can lead to an integrated intelligent inventory system exclusively characterised by its graphical user interface, data manager, pattern identifier, rule-based model selector, order quantity calculator and interpreter to supply users assistant information. The detailed functions of the proposed system components are described in the remainder of this section.

3.3.1 The User Interface

The user interface is the communication channel between the user and the system. It is vital that this communication should be as meaningful and friendly as possible because

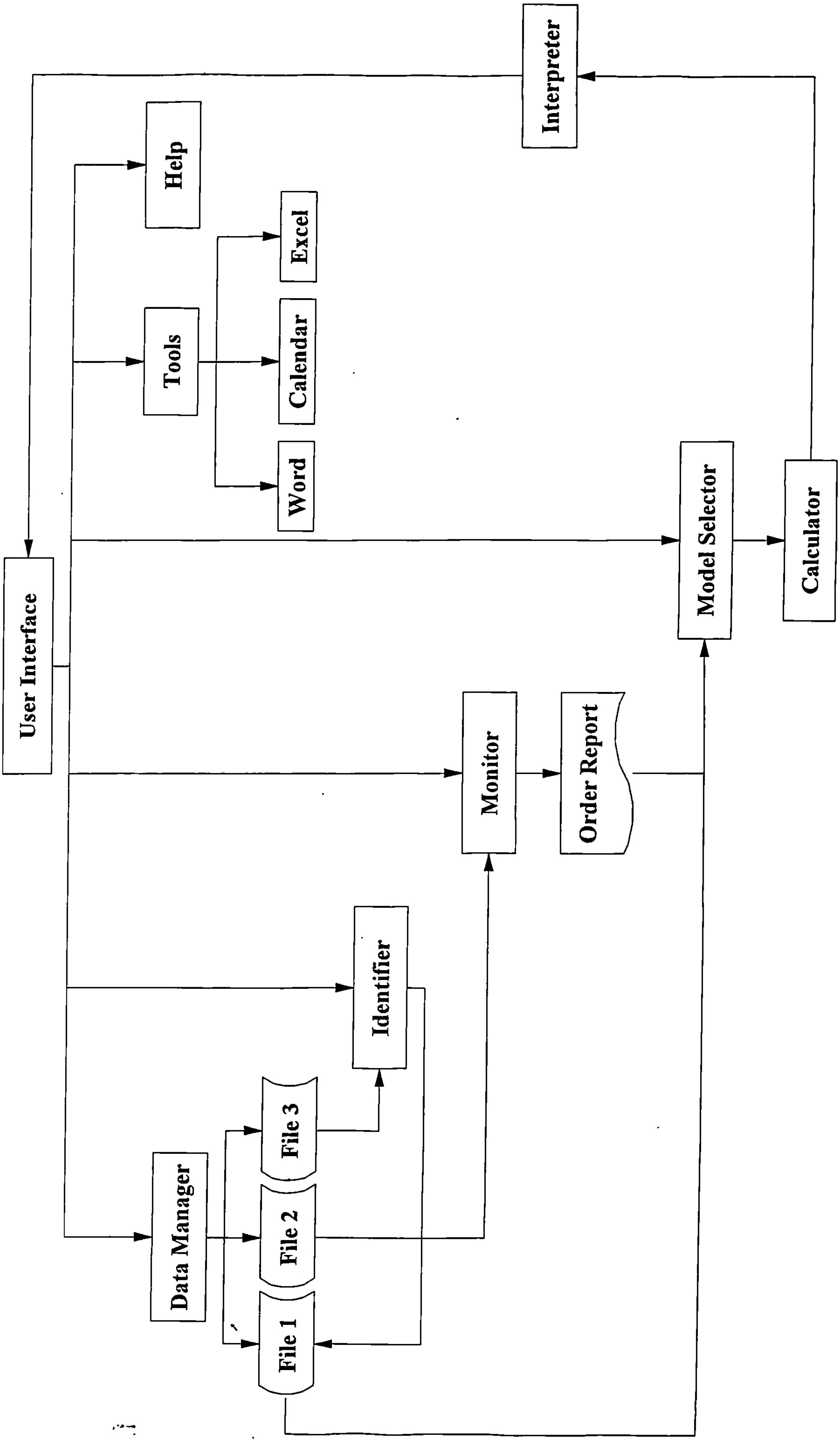


Figure 3.1 Outline Structure of the Proposed Inventory KBS

the user interface has a large role in judging the degree of usefulness of the system. If interface of the system is appropriately designed, the user then will be able to communicate effectively with the system.

The user interface of the system developed in this study includes a top-level menu (Fig. 3.2), the dialogue boxes, alert text, confirmation text, and help information facilities. The appearance of the user interface of this system is highly graphical and the menus, commands, and dialogue boxes are visually the same as other Windows applications.

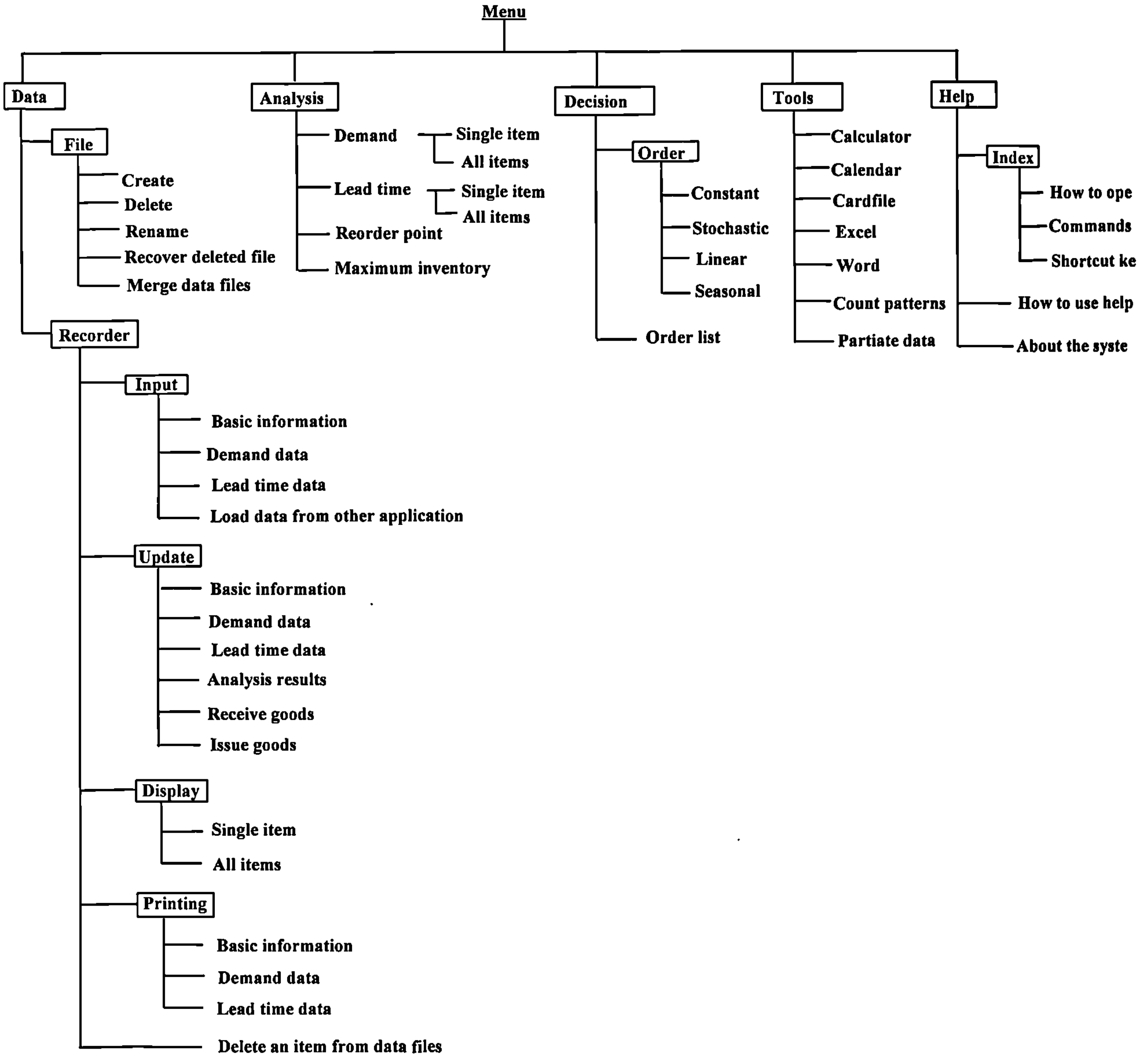


Figure 3.2 Top-Level Menu of the User Interface

The top-level menu gives the main options to operate the system. The dialogue box allows the user to respond to the system using the keyboard or a mouse. The alert text warns the user when unexpected information is occasionally input by the user. The confirmation text asks the user to confirm to continue the action when the action will lead to an unrecoverable result.

3.3.2 The Data Manager

This module is designed to manipulate the data files (Fig. 3.3) that contain all the data required for selecting suitable models and calculating order quantities. The operations of the data manager are classified into two categories. The first category of commands performs the general operations on data files such as creating a data file, deleting a data file, and renaming a data file. The second category of commands manipulates the records of the data files by carrying out the following actions:

1. Add a new item. This function allows the user to input the physical description information of a new item which includes code of product, name, unit, price, ordering cost, and holding cost.

2. Modify an existing item. After all the data are entered to the system, the user can modify them. When the product code of an item is given, the system will display the item's data stored in the system. After the user has completed the modification, the new messages will be stored into the corespondent file.

3. Delete an existing item. Items can easily be removed from the data file when it is not stored anymore.

4. Display an item. The system can consequently or randomly display the data of inventory items.

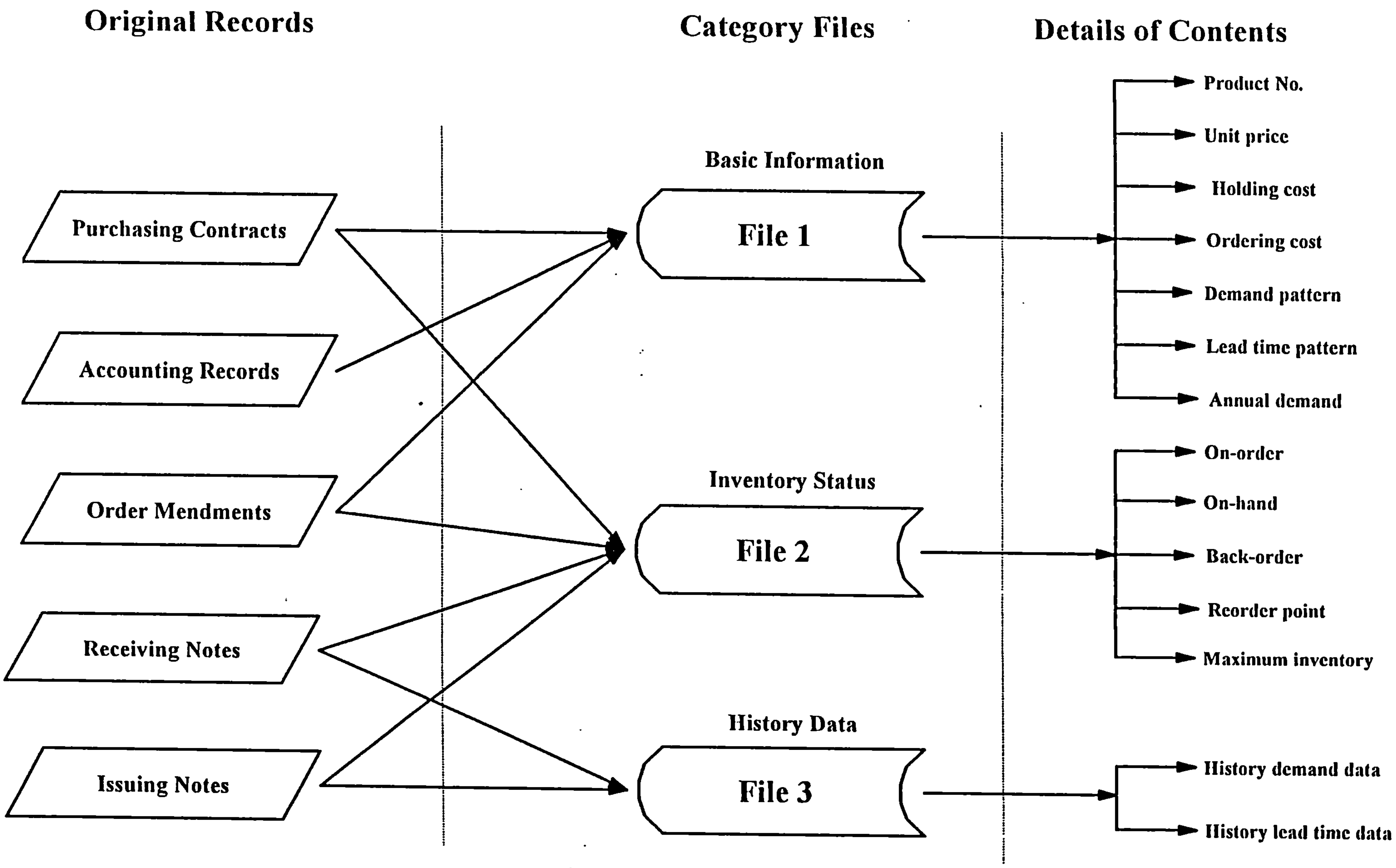


Figure 3.3 Structure of the Data Files

In additional, the data manager is also responsible for updating the usage data when new usage data becomes available. This procedure is illustrated in Fig. 3.4. The old usage data of an inventory item is stored in a file called historical data. The new usage data is accumulated from the receipts of issuing items from the warehouse. To reflect the new trend of demand of an item, the new usage data of an item will be put into its data record and the oldest usage data will be shifted out of the data record.

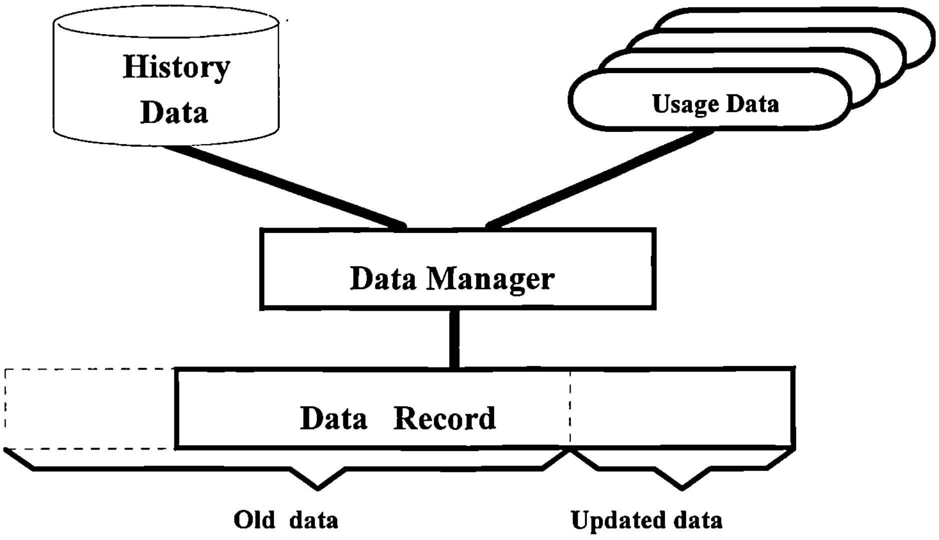


Figure 3.4 The Updating Procedure

3.3.3 The Pattern Identifier

It analyses the historical data to determine both the demand and lead time patterns which play a vital role in the process of selecting a suitable model in addition to other parameters such as unit price, supplying rate, etc. The results of the pattern identifier are stored as facts in the fact base (Fig. 3.5). The design and function of this module are described in Chapter 4.

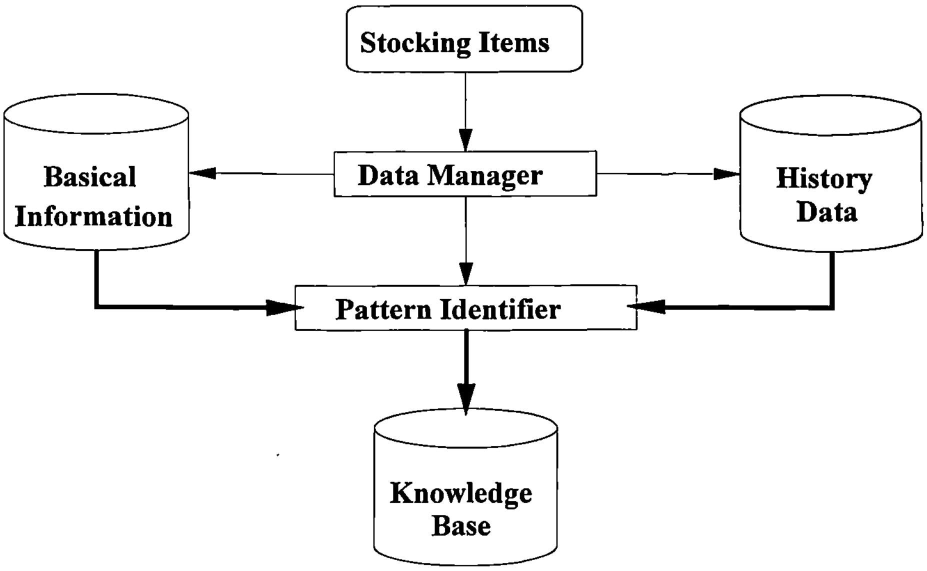


Figure 3.5 The Process of Identifying Demand Patterns

3.3.4 The Model Selector

It is a rule-based inference engine which can suggest an optimal or reasonable model according to the facts generated by the pattern identifier and the data provided to the system which describe the practical situation of an inventory item. The models that the system can use in the various situations are detailed in Chapter 5.

3.3.5 Calculator

The calculator is used to calculate reorder quantity in terms of the selected model and costs data stored in the files.

3.3.6 Inventory Monitor

This module updates the inventory position when items are issued (decreasing inventory) or received (increasing inventory). It can provide the current status of the inventory of any item and list all items that require replenishment.

3.3.7 Interpreter

It displays the final results and explains how and why the results were reached. The user of the system is thus able to gain access to the reasoning process behind the action of the system. This capability is illustrated by the following example.

Rule 1:

If DEMAND = constant
and LEAD-TIME = constant
Then MODEL-TYPE is EOQ.

Rule 2:

If MODEL-TYPE is EOQ
and SHORTAGE = allowed
Then MODEL = EOQ with shortage allowed.

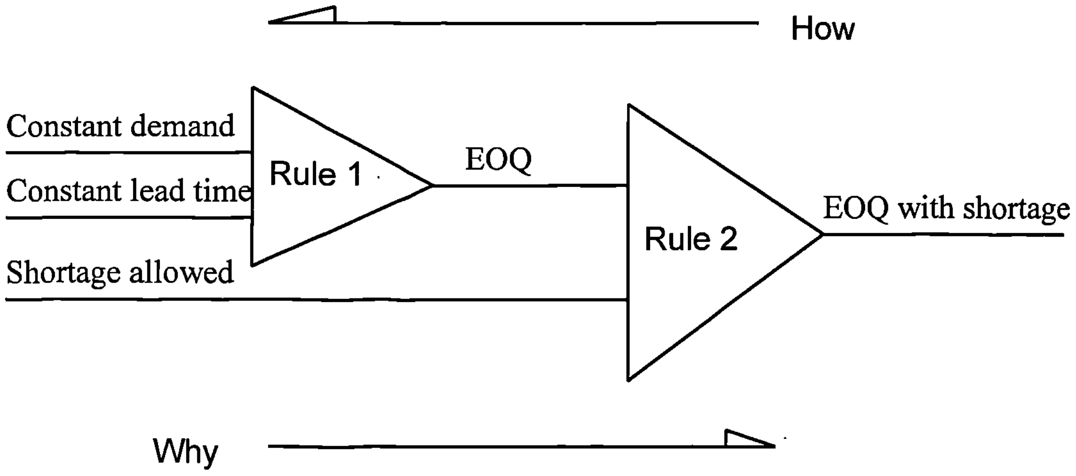


Figure 3.6 Explanation of Reasoning Process

The reasoning process of the above rules is explained in Fig. 3.6 and the interpreter will supply an explanation message about the process to the user. First, if both the demand

and lead time are constant, then rule 1 is triggered and an EOQ model is selected. If the EOQ is selected and shortage is allowed, then the rule 2 is triggered that leads to a final decision to use a modified EOQ which can deal with the situation in which shortage is allowed. In Fig. 3.6, the Why direction displays the forward-chain reasoning process, and the How direction explains how the goal is achieved.

3.3.8 Tools

These are a number of commercial software which have been integrated into this system and used to carry out the various operations and enhance the functions of the system. They are:

Calendar which is an electronic calendar.

Cardfile which is an electronic data and information organiser.

Excel provides the automated business tools needed for data analysis, list keeping, calculations, and presentation tool for reporting results.

Microsoft Word which is a word processor and is used to produce periodical reports.

3.3.9 Help

It provides the information to tell the user how to operate the system and explains the functionality of commands and shortcut keys. The help menu contains three sections (see Appendix A). The first section explains how to perform every operation step by step. The second section explains the precise meaning of the commands because they are very briefly indicated in the menus. The third section concerns with using keyboard to operate the system. The shortcut keys will help the user to highlight a menu, select a command, and execute a procedure.

3.4 The Salient Features of the Proposed System

Compared with other systems published in literature (see Section 2.5), this system has a variety of advanced features to aid the inventory managers in making decisions. We will overview these advanced topics under the following categories:

1. A Pattern Identifier has been designed and integrated into the system to estimate parameters related to model selection. The pattern identifier consists of various statistical methods for the analysis of the usage data of an inventory item to automatically estimate the parameters for selecting the most appropriate model to use. The details of the mathematical methods and the identification procedures are discussed in Chapter 4.

2. The knowledge base has been extended to deal with various special situations of EOQ model in addition to stochastic, slow-moving, linear and seasonal demands. The inventory models embedded in this system are detailed in Chapter 5.

3. A Data Manager has also been developed to manipulate the data required by the pattern identifier and the information used to select a suitable model and calculate an order quantity. The development of the data manager is discussed in Chapter 6.

4. The business world over the past decades has stored tremendous amounts of information in computer-based database files and spreadsheet files. As discussed in Section 3.3.7, this system is designed to access other commercial software such as Excel which is one kind of spreadsheet application and is widely used by the business world. In theory, this system can interface with any MIS used in inventory management because the developing tool, Visual Basic for Windows, can call any DOS based executive program and any application for Microsoft Windows from within its

environment. The method of how to access to other applications will be detailed in Chapter 6.

5. The development of this system is based on practical situations and also on co-operation with international manufacturing and service organisations. The testing and verification processes are carried out using the real data provided by the co-operating companies. Chapter 7 will discuss the outcomes of the system in terms of the practical situation.

This chapter briefly discussed the outline structure and the distinct features of the proposed system. The most important two components, the pattern identifier and the model base, comprise the core of the system because they determine the system's functions and behaviour. The following two chapters will discuss in detail the pattern identifier and the model base respectively.

Chapter 4

DEMAND PATTERN IDENTIFICATION

This chapter is concerned with the demand pattern identifier of the intelligent inventory system. The pattern identifier has the ability to identify constant demand, estimate a probability distribution, regress trend demand, and recognise seasonal movement. The first section discusses the importance and vital role of the pattern identification process in the operation of an intelligent inventory system. Following the classification of demand patterns in the second section, the third section is devoted to the process of demand identification.

4.1 The Importance of Demand Pattern Identification

The selection of policy and hence a model that will be employed to achieve successful inventory management using the KBS is based on the current values of parameters which define the state of the inventory item. Some of the parameters such as the discount break points and the foreign exchange rate can be highly changeable or difficult to predict using mathematical methods, but they can be readily known to the inventory managers. Other parameters such as demand and lead time patterns are usually difficult to be identified without carrying out statistical analysis of the historical data.

As the time passes, we may find that the future demand pattern of an item may differ from its historical pattern, and in consequence may conclude that the previous adopted model is no longer appropriate given the current values of the item's parameters. As more transaction records become available, the identified demand pattern of an item must be checked and updated if required.

The above discussion clearly indicates that in order to optimise the inventory operations of a company, the expert system should be able to compute and update the optimum values of parameters for thousands of the stocked items and automatically select and update the selection of suitable models for each item. Therefore, a vital step in establishing an intelligent inventory decision support system is to develop an interface which is able to identify the demand patterns of inventory items by analysing the historical usage records. Once the demand pattern has been identified, it becomes the basis for the subsequent selection of a suitable replenishment model. The alternative to this would be to leave the decision of model selection to an analyst familiar with the statistical forecasting methods and having an overall knowledge of the existing inventory models. The costs of such highly specialised analysts will most likely offset the savings resulting from the extra efforts spent on selecting the most suitable inventory model for each of the item.

There are many good statistically based forecasting models which generate forecasts by extrapolating patterns of historical data [121, 122]. The general methods used to predict demand patterns of stocked items have been discussed by Plossl [5], Peterson and Silver [36], and Lewis [123]. This chapter will discuss the statistical forecasting methods used for identifying demand patterns based on a demand pattern classification proposed in this thesis.

4.2 Classification of Demand Patterns

The nature of the demand for an inventory item can be stable or highly variable. As discussed in Chapter 1, the independent demand is influenced by market forces or failure patterns, in the spare parts case, rather than by failures related to operations. Many published papers deal with the classification of inventory models rather than demand patterns (see Section 2.1). Based on the study of the nature of the demand of inventory items, a proposed classification of demand patterns of inventory items is given in Fig. 4.1 [124].

Muriall [124] proposed that all inventory items with independent demand can be divided into statistically unpredictable and statistically predictable patterns. We propose the addition of a third category, i.e. the low demand (or slow-moving items). In what follows, we describe each of these demand patterns.

4.2.1 Statistically Predictable Demand Patterns

The statistically predictable demand patterns have relatively smooth and repetitive fluctuations and can be analysed using statistical forecasting methods. The demand of these patterns may change with time, and hence they are further divided into time dependent and time independent demand patterns.

In the time dependent category, the type of replenishment quantity is dependent on the time when the replenishment decision is made, while it is not the case in the latter.

Under the condition of time independent demand, uncertainty may exist. If the demand is reasonably stable and can be accurately forecasted, then it can be assumed to be constant. In other case, if there is significant uncertainty which can be specified by a theoretical or empirical distribution, the demand is referred to as stochastic demand.

Any error in demand estimate will result in variance from the optimal order quantity, so that it is reasonable to divide constant demand into absolutely constant demand (variation is near to zero) and quasi constant demand (variation is less than a given constant).

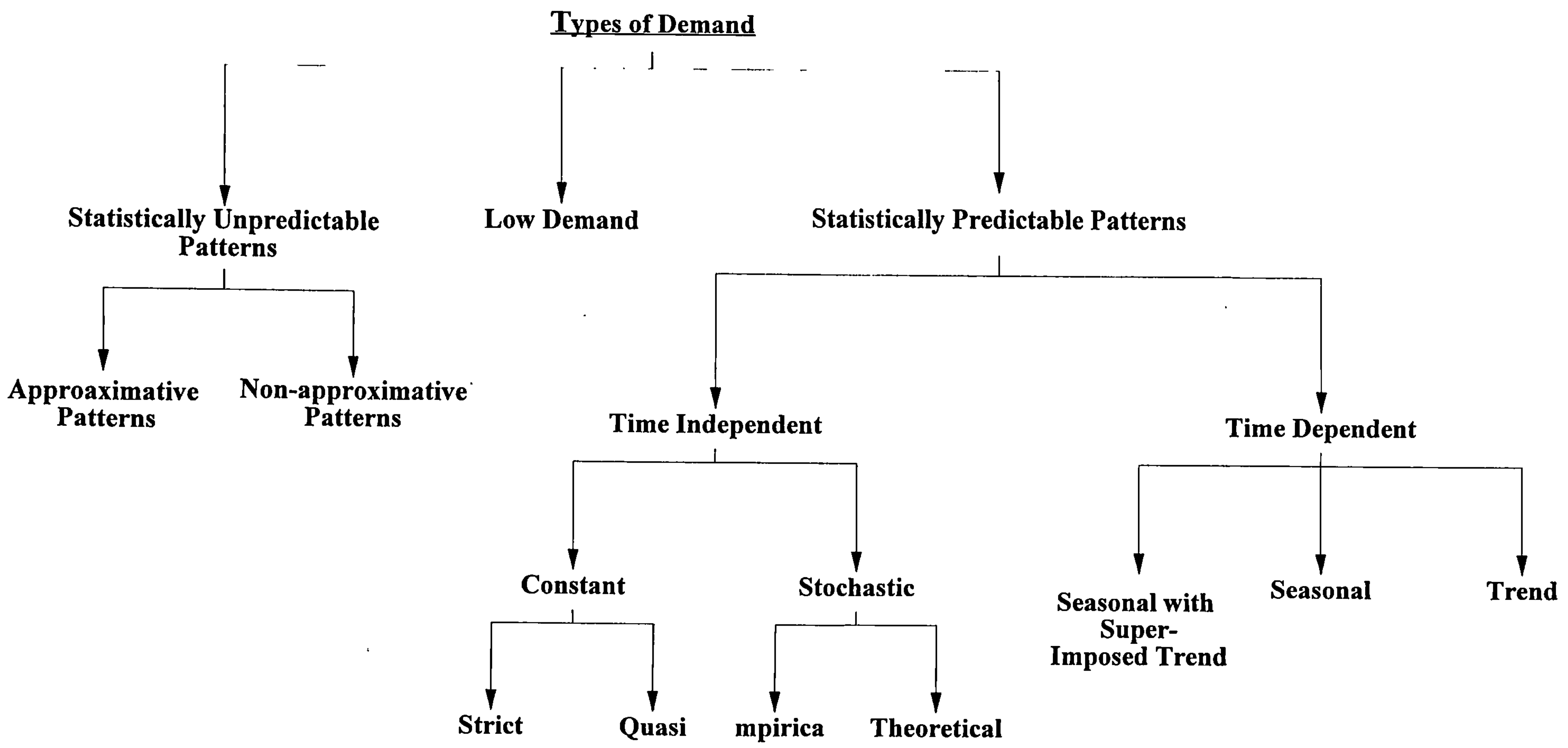


Figure 4.1 The Classification of Demand Patterns

Probabilistic demand may follow some kinds of theoretical probability distribution or empirical distribution. In reality, all of the theoretical distributions of demand should be discrete. Classifying them into discrete and continuous distributions is due to the fact that some continuous distribution can asymptotically describe some demand patterns.

The probability distributions that the system can identify are the Poisson and Normal distributions. The Poisson probability distribution often provides a good model for the probability distribution of rare events that occur infrequently in space, time, volume, or any dimension. Although the normal distribution is completely specified by its mathematical rule and it is not a fact of nature that one actually observes to be exactly true [125], it is the fact that measurements on many random variables appear to have been generated form population frequency distribution that are closely approximated by a normal probability distribution. If the demand does not fit either the discrete or the continuous distributions which the system can identify, then the system will suggest a random demand and estimate the mean and variance of the demand. It is easy to extend the system to identify more theoretical distributions by adding appropriate submodules.

4.2.2 Time Dependent Demand Pattern

Time dependency of demand may be caused by seasonal variation which can also mask the trend in the original data. If a product is at its growth or saturation stage of life cycle, its demand will tend to increase or decrease respectively. Therefore, we classify the time dependent demand patterns into trend, seasonal, and seasonal with super imposed trend.

4.2.3 Statistically Unpredictable Patterns

The statistically unpredictable demand patterns have sudden fluctuations caused by lumpy demand. They can be classified into approximative patterns and non-approximative patterns. The demand of an item with non-approximative pattern has severe, but regular, highs and lows of demand which do not reoccur at the same time each year but at predictable intervals. The demand of an item with approximative pattern has severe and random highs and lows of demand which do not reoccur predictably. Conventional forecasting techniques, including statistical techniques, are not suitable for dealing with these demand patterns. However, they may be approximated by a statistically predictable pattern (see Section 7.4.1) within tolerable range of errors.

4.2.4 Low Demand

An inventory item which has a low demand (or slow-moving item) means that there are very few transactions occurring over a reasonable historical period. The definition of low demand pattern (or a slow-moving item) is arbitrary. For example, in this study, if an item's annual demand is six or less (or less than a given constant) and the demand at each time is one or zero, it is defined as an item with low demand pattern (or a slow-moving item). Forecasting the demand of an item with low demand pattern is not an easy matter. Purely an objective estimate of a demand rate is usually not feasible. Instead, we must take advantage of the subjective knowledge of inventory managers.

4.3 Test Process

To identify the demand patterns described in the previous section, the system should have the ability to separate seasonal movements from the basic demand, if there is any, and then identify the basic demand pattern. For the non-seasonal demand, the system

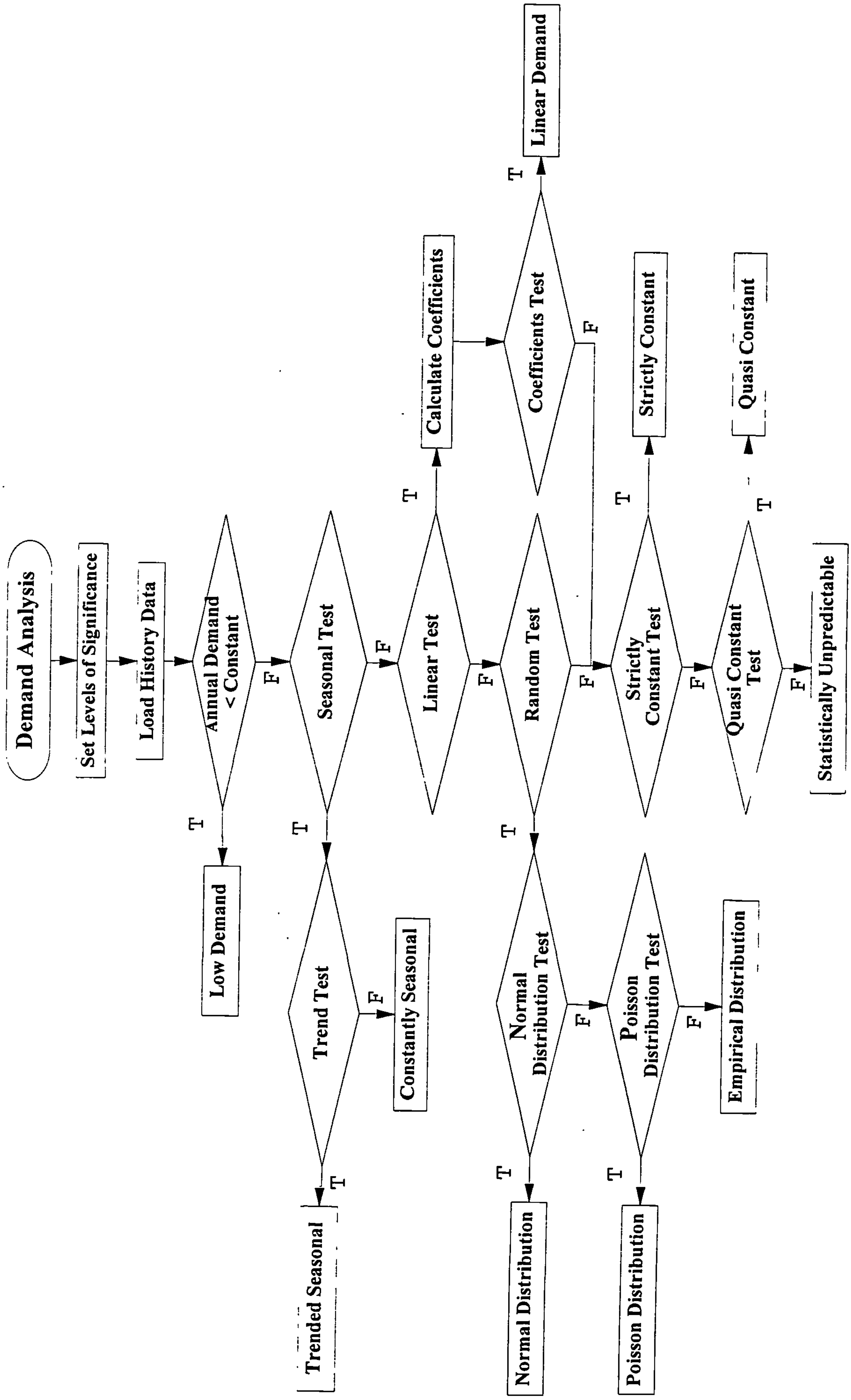


Figure 4.2 The Flowchart for Demand Pattern Analysis

should be able to identify the stationary, linear or probabilistic demand patterns. Fig. 4.2 shows self explanatory flow chart for the demand pattern analysis process which consists of the following steps:

1. Enter the (or accept the system's default) significance levels for each test method which are discussed later and summarised in Fig. 4.4;

2. Load history demand data into the pattern identification module;

3. Identify the items with low demand pattern, if the annual demand of the item is less than a given constant specified by the user (or inventory manager), the tested item has a low demand. Otherwise, the system continue the demand identification process;

4. Test whether there are seasonal movements in demand. If the demand is seasonal, test whether the seasonal movements are based on linear trend or constant demand. Otherwise, go to step 5;

5. If the demand has no seasonal movements, test whether it has a linear trend, if the demand has a linear trend, then check that the linear coefficient is significantly different from zero. If true then the demand has a linear trend. Otherwise go to step 7. If the demand has no trend go to step 6;

6. The demand has no trend or seasonal movements. Test whether the demand is random If true select an appropriate distribution , first by testing Normal distribution assumption, if true the demand is normally distributed. If false test for the Poisson distribution, if true the demand follows Poisson distribution. Otherwise use empirical distribution. If the random test is false go to step 7;

7. The final tests are to decide if the demand is strictly constant, quasi constant, or statistically unpredictable. If the variance of the demand is zero, the demand pattern is strictly constant. If the variance of the demand is greater than zero but less than a specified constant, the item has quasi constant demand.

8. Finally, if the item's demand has failed all of the above tests, it is defined to have statistically unpredictable demand pattern.

The statistical methods used for identifying different demand patterns are detailed below.

4.3.1 Seasonal Movements Test

Several methods such as Holt-Winters model, seasonal exponential smoothing, are often used to forecast seasonal time series. However, the pre-condition of applying these methods is that we have recognised that the time series has seasonal movements. The methods themselves do not have the statistical ability to identify whether a time series has seasonal movements. Therefore, it is not suitable to use this method to identify if there are seasonal movements in a time series. In this study, the Kruskal-Wallis Test [121] is adopted for detecting seasonal movements, which is based on testing the rank of specific seasonal $(Y_i'(t))$ which is calculated by the following steps:

1. Calculate L-period moving totals of the original data;
2. Calculate 2-period moving totals of the moving totals resulting from step 1;
3. Divide the result of step 2 by $2L$, to obtain the centred moving average CMA;
4. Subtract CMA from original data, that produces series of specific seasons, i.e.

$$Y'(t) = Y(t) - CMA(t). \tag{4-1}$$

Hypotheses

H_0: the demand has no seasonality ($S_1 = S_2 = \dots\dots = S_L = 0$).

H_a: the demand has seasonality ($S_{j \neq} 0$).

Test statistic

$$H = \frac{12}{n(n+1)}[\sum \frac{R_i^2}{n_i}] - 3(n+1) \tag{4-2}$$

Where

L= the number of seasons.

n_i = number of observations in *ith* season.

n = total number of specific seasons (= $\sum n_i$).

Y_t' = specific seasonal for time t.

$R_i = \sum Rank(Y_t')$ *ith* season.

Decision Rules

If ($H > \chi^2_\alpha(L-1)$) reject H_O, otherwise, accept H_O. Where $\chi^2_\alpha(L-1)$ is Chi-square distribution with L-1 degrees of freedom.

4.3.2 Trend Test

The non-parametric test for trend can be constructed using Spearman's rank-order correlation coefficient r_s It is defined as Pearson's product moment correlation coefficient r between the ranks of two variables t_i and y_i, i.e. to replace a sample of pairs of measurements (t_1,y_1), (t_2,y_2), ... (t_n,y_n) with their respective pairs of ranks $(R(t_i),R(y_i))$ [123]. The testing process is explained as follows.

Hypotheses

H_O: the demand has no trend.

H_a: the demand has (upward or downward) trend.

Test Statistic

- Small Sample ($n \leq 30$)

If there are no ties (two or more observations exactly equal to each other):

$$r_s = 1 - \frac{6\sum_{i=1}^{n}(R(t_i)-R(y_i))^2}{n(n^2-1)} \tag{4-3}$$

and when there are ties, then the exact relation is [126]:

$$r_s = \frac{1-\frac{6}{n^3-n}\left(\sum_{i=1}^{n}(R(t_i)-R(y_i))^2+\frac{1}{2}\sum_{j=1}^{m}(f_j^3-f_j)+\frac{1}{2}\sum_{j=1}^{k}(g_j^3-g_j)\right)}{\left(1-\frac{\sum_{j=1}^{m}(f_j^3-f_j)}{n^3-n}\right)\left(1-\frac{\sum_{j=1}^{k}(g_j^3-g_j)}{n^3-n}\right)}. \qquad (4\text{-}4)$$

where

f_i = the number of ties in the *ith* group of ties among the $R(t_i)$'s;

g_i = the number of ties in the *ith* group of ties among the $R(y_i)$'s;

m = the number of groups of ties among the $R(t_i)$'s;

k = the number of groups of ties among the $R(y_i)$'s.

- Large Sample ($n > 30$)

$$Z = \frac{r_s - \mu_{r_s}}{\sigma_{r_s}}, \quad \text{where } \mu_{r_s} = 0, \text{ and } \sigma_{r_s} = \frac{1}{\sqrt{n-1}} \qquad (4\text{-}5)$$

Decision Rules

- Small Sample

If ($| r_s | > r_{\alpha/2}$) reject H_0, otherwise accept H_0.

- Large Sample

If ($| Z | > Z_{\alpha/2}$) reject H_0, otherwise, accept H_0.

4.3.3 Test for Randomness

The turning points test [121] is used to check whether the demand pattern is random or not. The turning point in a time series is a point where the series changes direction. Each such point represents either a local "peak" ($X_{t-1} < X_t \geq X_{t+1}$ or $X_{t-1} \leq X_t > X_{t+1}$) or a local "trough" ($X_{t-1} \geq X_t < X_{t+1}$ or $X_{t-1} > X_t \leq X_{t+1}$) in the series. For example , the underlined numbers in the following series represent the turning points of the series.

$$13, \underline{29}, \underline{19}, \underline{31}, \underline{20}, 15, \underline{40}, 36, 29, \underline{19}, 41.$$

This method is based on the premise that a trended or positively autocorrelated series should have fewer turning points than a random one and a negatively autocorrelated series should have more. If the series is actually a random series, the sampling distribution of the number of turning points U is approximately normal for even moderate number of observations ($n \geq 10$).

Hypotheses

H_0: the demand is random.

H_a: the demand is not random.

Test Statistic

Moderate or Large Sample($n \geq 10$)

$$Z = \left| \frac{U - \mu_u}{\sigma_u} \right| \qquad (4\text{-}6)$$

where U is the number of turning points of n observations in a series and

$$\mu_u = \frac{2(n-2)}{3}, \qquad \sigma_u = \sqrt{\frac{16n-29}{90}}.$$

Decision Rules

If (| Z | > $Z_{\alpha/2}$) reject H_0, otherwise, accept H_0.

4.3.4 Identification of Probability Distribution

Under stochastic condition, we need to identify a suitable probability distribution for demand. The process shown in Fig. 4.3 is designed to identify the probability distribution when the demand is probabilistic.

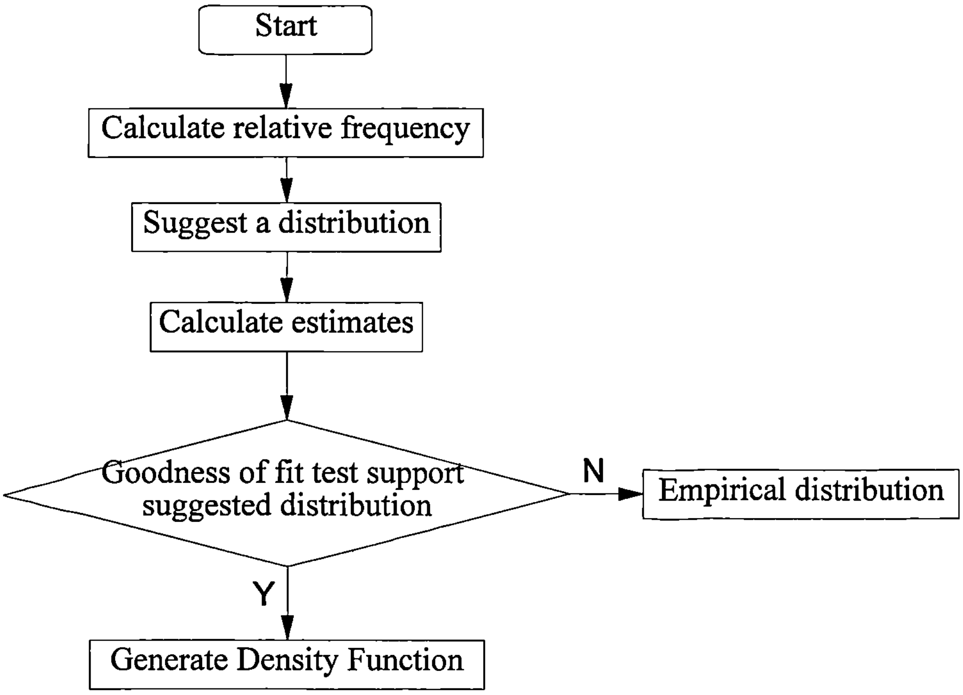

Start
Calculate relative frequency
Suggest a distribution
Calculate estimates
Goodness of fit test support suggested distribution
N
Empirical distribution
Y
Generate Density Function

Figure 4.3 Identification of Probability Distribution

As explained before two theoretical distributions can be identified by the current system, i.e. the normal distribution and Poisson distribution. If the demand is random and tests for Normal and Poisson distributions were rejected, then empirical distribution is used instead.

To test the appropriateness of using the Poisson distribution to describe the demand pattern the following methods are used to test the goodness of fit of the probability distribution.

Test Statistic

- Large Sample($n > 30$)

For the cases of large samples, the modified Pearson's chi-square [127] statistic is used, i.e.

$$\chi^2 = \sum_{i=1}^{m} \frac{[f_i - n\pi_i(\hat{\theta})]^2}{n\pi_i(\hat{\theta})}. \tag{4-7}$$

- Small Sample ($n \leq 30$)

For the cases of small samples, the statistic can be expressed as [127]

$$\chi^2 = \frac{1}{n}\sum_{j=1}^{m} \frac{f_j^2}{\pi_j(\hat{\theta})} - n. \tag{4-8}$$

Where θ stands for the probability distribution parameter to be tested, and $\hat{\theta}$ is an estimate of θ obtained from the sample, $\pi_i(\theta)$ is the probability of category *i* if the assumption that $\hat{\theta} = \theta$ is true and *m* is the number of categories of sample which is obtained using a rule of thumb, say, the sample size n is four or five times *m*.

Decision Rules

- Large Sample

If $\chi^2 > \chi_{\alpha/2}$, then reject suggested distribution, otherwise accept suggested distribution

- Small Sample

If $\chi^2 > \chi_{\alpha/2}$, then reject suggested distribution, otherwise accept suggested, distribution.

If the demand has a Poisson distribution, the average (mean) equals to its variance. But this property of Poisson distribution can not be used to test whether the demand series has a Poisson distribution. Obviously other distributions may also have the property that the mean is equal to its variance in particular case, e.g. a Normal distribution with its mean = variance = 1.

To test the appropriateness of using the Normal distribution to describe the demand pattern the Kolmogorov-Smirnov statistic D_n [127] is used:

$$D_n = \sup_{all\ x} |F_n(x) - F(x)| \qquad (4\text{-}9)$$

Where *F(x)* is the suggested distribution function and $F_n(x)$ is sample cumulative frequency.

If $D_n > K$ (reject limit) reject suggested distribution, otherwise , accept suggested distribution.

Unfortunately, a closed-form expression for the integral of some density functions, including the normal distribution, $\int_a^b \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$, does not exit, and hence its evaluation can be obtained by approximate procedures based on the Trapezoidal rule, i.e. to use Newton-Cotes formula

$$\int_{x_1}^{x_2} f(x)dx = h(\frac{1}{2}f_1 + \frac{1}{2}f_2) + O(h^3 f''). \qquad (4\text{-}10)$$

By adding the outcomes of the *N-1* integration in the intervals (x_1,x_2), (x_2,x_3), ... (x_{N-1},x_N), we obtain an extended formula for the integral from x_1 to x_N

$$\int_{x_1}^{x_N} f(x)dx = h(\frac{1}{2}f_1 + f_2 + ...+ f_{N-1} + \frac{1}{2}f_N) + O(h^3 f'' / N^2) \qquad (4\text{-}11)$$

which is called the extended Trapezoidal rule [126]. The most course implementation of the Trapezoidal rule is to average the function at its endpoints *a* and b. The first stage of refinement is to add this average to the value of the function at the halfway point. The second stage of refinement is to add the values at the 1/4 and 3/4 points. And so on. This refinement process is controlled by comparing the difference between the old integral value and refined integral value with a present error limit.

4.3.5 Linear Model Test

If demand is found to be increasing (or decreasing), then the parameters of the linear statistical model:

$$Y = \beta_0 + \beta_1 X \tag{4-12}$$

are estimated using the least square methods [127], i.e.

$$\hat{\beta}_1 = \frac{n(\sum(xy)) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \tag{4-13}$$

and

$$\hat{\beta}_0 = \frac{\sum y - \beta_1(\sum x)}{n} \tag{4-14}$$

The significance of a non-zero value of β_1 is tested by the following procedure.

Hypotheses

H_0: the testing parameter $\beta_1 = 0$.

H_a: the testing parameter $\beta_1 \neq 0$.

Test Statistic

$$T = \frac{\hat{\beta}_1 \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})}}{\sqrt{\frac{\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2}{n-2}}}, \tag{4-15}$$

Which is distributed approximately as student's distribution with *n-2* degrees of freedom.

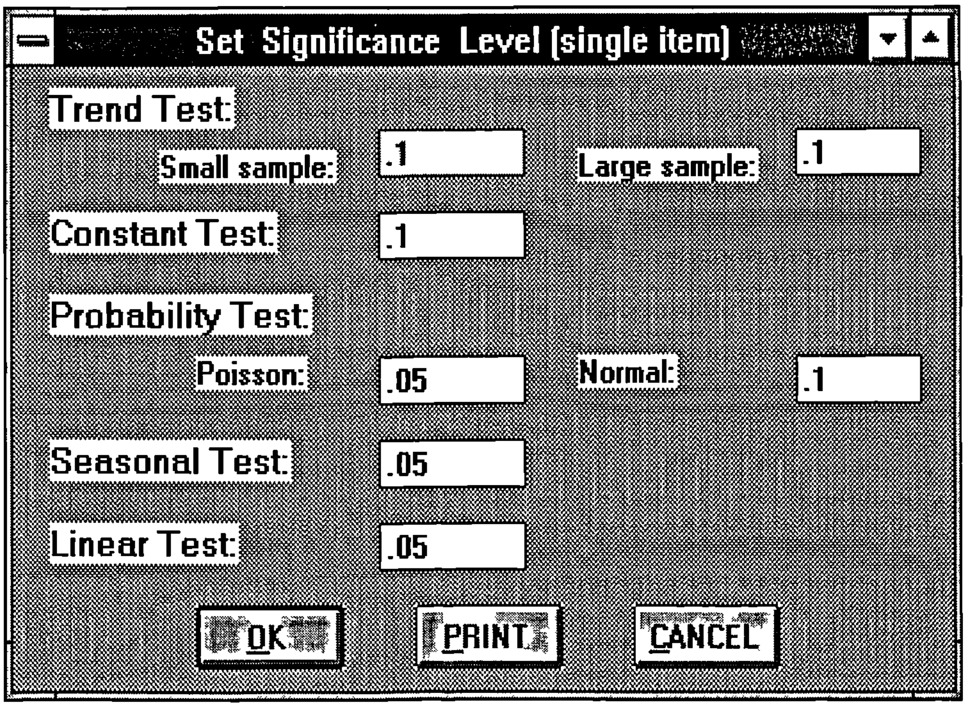
Decision rules

If $T > t_{\alpha/2}$ reject H_0, otherwise, accept H_0.

4.3.6 General Comments on Testing Procedure

The procedure for identifying an item's demand pattern always starts with selecting significant levels for each testing method (Fig. 4.4). Instead of accepting the system's default values illustrated in Fig. 4.4, The user can also specify these significance levels in the following values for each of the test methods:

0.005, 0.010, 0.025, 0.050, 0.100.



Set Significance Level (single item)

Trend Test:

Small sample: .1 Large sample: .1

Constant Test: .1

Probability Test:

Poisson: .05 Normal: .1

Seasonal Test: .05

Linear Test: .05

Figure 4.4 System Default Significance Levels

In selecting the methods for the statistical tests discussed above, the following considerations have been taken into account. First, nonparametric methods are most preferred because we can not assume what kind of demand that an item has before the testing was carried out. Secondly, the methods which have less assumption are selected because more restrictions tend to limit the applicability of the methods. Finally, the selected methods must be suitable for programming because they will be incorporated into a computerised system.

During the above testing process, the average (or mean) and variance of tested item's demand are estimated as follows [125]:

Average demand

$$\overline{d(i)} = \frac{1}{n}\sum_{i=1}^{n} d(i)$$

and variance of demand

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d(i) - \overline{d(i)})^2} .$$

When some new demand data are available, some the oldest demand data will be deleted and the average and variance of demand will be estimated with the new set of demand data. The details of how often to repeat the demand pattern identification and the impact of changing significance levels on the outputs of analysis will been discussed in Chapter 7.

4.4 Identifying the Lead Time Pattern

Most inventory items usually have a lead time between placing an order and receiving the ordered quantity. The lead time is caused by the time spent on transportation and/or the time spent on manufacturing ordered items. It is reasonable to classify the patterns of lead time into constant or stochastic. The methods used to identify lead time patterns depend on the kind of the available lead time data. If the historical data of lead time were recorded, then statistical methods can be used to identify its pattern. Otherwise, a subjective estimation method can be adopted to obtain the distributions of lead time.

4.4.1 Statistical Methods

In general the procedure of identifying lead time patterns is similar to the process of identifying demand patterns. However, in practice this can be simplified because the possible lead time patterns are not as complicated as demand patterns.

The runs test is first used to test whether the lead time is constant or probabilistic. If the lead time is constant, then its value can be estimated by the average. Otherwise, the Kolmogorov-Smirnov statistic is used to test the goodness of fit of the data to the Normal distribution and the modified Pearson's chi-square statistic which is used to test the goodness of fit of the data to Poisson distribution. If the data of lead time does not fit either of these two distributions, then an empirical distribution can be used.

4.4.2 Subjective Estimation of Lead Time Distribution

During this study, many visits have been made to some industrial organisations. However, some companies do not record the lead time data. In order to obtain the lead time information which used to make order decisions, a subjective estimation should be applied. The method of subjective estimation adopted in this study is based on the Bayes theorem. Suppose that the range of lead time is divided into *m* possible intervals, and there are *E* experts who can estimate the number of lead times that falls in each interval. From the property of the Dirichlet distribution the consensus estimation of the frequency of lead time in the *ith* interval, n_{id} can be obtained as follows [128]:

$$n_{id} = \frac{\sum_{e=1}^{E} w_e n_{ie}}{\sum_{i=1}^{m}\sum_{e=1}^{E} w_e n_{ie}} \sum_{i=1}^{m} n_{ie} \qquad (4\text{-}13)$$

where

$i = 1, 2, ..., \text{m}$,

E = the number of experts,

m = the number of intervals defined in the required histogram of the lead time,

n_{ie} = the subjectively estimated number of the lead times in the *ith* interval specified by expert e, $i = 1, ..., m$, and $e = 1, ..., E$,

n_{id} = the consensus estimation of the number of the lead times in the *ith* interval, $i = 1, ..., m$,

w_e = the weight determined from the prior belief about the individual expert.

For convenience we can set the bounds $\sum_{e=1}^{E} w_e = 1$, i.e. the sum of weights given to all expert = 1 and that each w_e can be obtained by using equal weights or weights proportional to the experts' expertise. Here we propose to use the latter since each expert may have different expertise on the distribution of lead time. However, this method usually causes underestimation of the lead time.

4.5 Summary

The demand pattern identification module serves as an interface to the intelligent inventory decision support system. It combines logical and statistical methods to identify a pattern for the data of each independent demand inventory item, from the historical data without the need to interrogate the user. The day to day transaction data are feedback into this interface to update and / or identify the future demand pattern. The output of this module is essential in selecting a suitable inventory model. With this module, a user can analyse demand patterns without much knowledge of statistics. But a user who has some statistical knowledge can run it more effectively by adjusting the test criteria instead of accepting the default values (see Section 7.4.3).

By using the system to identify the demand patterns of inventory items, the user need to specify the length of season, the criterion of annual demand for slow-moving items, and the critical value to identify quasi constant demand (see Section 7.3).

Another important input that the user has to specify is the time interval of the successive observations of demand data (see section 7.3). It will be used to calculate annual demand for testing low demand items and to estimate the demand standard deviation during a lead time when the lead time demand is not presented in the same time interval as the demand data itself (see Section 5.3.2.2).

Although this module is primarily designed to identify the demand pattern, it is relatively independent and can be used in identifying patterns of other parameters including the lead time.

Chapter 5

INVENTORY MODELS EMBEDDED IN THE KNOWLEDGE BASE

In general, the practical inventory situations are complex. It is almost impossible to design a system that can handle every kind of inventory management situation. The proposed system in this study is designed to provide the user with advice on inventory situations with demand patterns that fall into one of the categories classified in the previous chapter. In this chapter the most significant inventory models related to each of these demand patterns are discussed and compared. Suitable models are then selected from the published literature but some models are specially developed for the proposed system if a gap is identified. Following the introduction, Section 5.2 is concerned with inventory models for constant demand. This is followed by the discussion of the models for probabilistic demand in Section 5.3, low demand in Section 5.4, linear demand in Section 5.5, and seasonal demand in Section 5.6. This chapter is concluded with a summary in Section 5.7.

5.1 Introduction

The kind of a model which is used to make an order decision and calculate an order quantity depends mainly on the item's demand and lead time patterns in addition to information about the various inventory costs, price discount, shortage cost, supply rate, etc. In this study, the models included in the system's knowledge base can deal with the inventory items that have the following properties [129]:

- **Demand patterns**

  1. The constant demand;

2. The probabilistic demand;
3. The low demand;
4. The linear trend demand;
5. The seasonal demand.

- **Lead time patterns**

1. The constant lead time;
2. The probabilistic lead time.

To build up the knowledge base of the proposed system, the models which deal with each inventory situation, primarily according to the identified demand pattern, are compared and the most suitable models have been selected for making reordering decisions and calculating the order quantity in accordance with the criteria of applicability, easiness of understanding, and being adaptable to a computer solution. Several models which are mainly concerned with situations of constant demand with discount, shortage, and restricted supply rate have been developed. Furthermore, the selected model for demand with linear trend has been modified to cope with the situation where the number of reorders is not an integer. Also a method to deal with seasonal demand, with basic demand element that is either constant or has a trend, has been proposed.

To describe the decision rules adopted in the proposed system it is useful to define the following stock levels as they play an important role in deciding the optimal order quantity:

a. On-hand inventory

This is the inventory that is physically on the shelf (or in the storeroom); it can never be negative. This quantity is relevant in determining whether a particular customer demand is satisfied directly from the shelf.

b. On-order inventory

This is the inventory which has been requisitioned but not yet received.

c. Backorder

It is an unfulfillment of customer order or commitment. It is an immediate (or over due) demand for an item that is out of stock. Backorders should occur only when customers allow shortage to be fulfilled later without the threat of taking their business elsewhere.

d. Safety inventory

The safety inventory is defined as the average net inventory just before a replenishment arrives. A positive safety inventory provides a buffer against larger-than-average lead time or larger-than-average demand during the replenishment lead time duration.

e. Net inventory

Net inventory is the difference between on-hand inventory and backorders, i.e.

Net inventory = (On-hand) - (Backorders)

This quantity can become negative. It is seen to be a component of safety stock.

5.2 Models for Constant Inventory Problems

The constant inventory problems are characterised by the following features:

- Both the demand and lead time patterns are known with certainty and stay constant in the future.
- The ordering cost, and holding cost are constant.

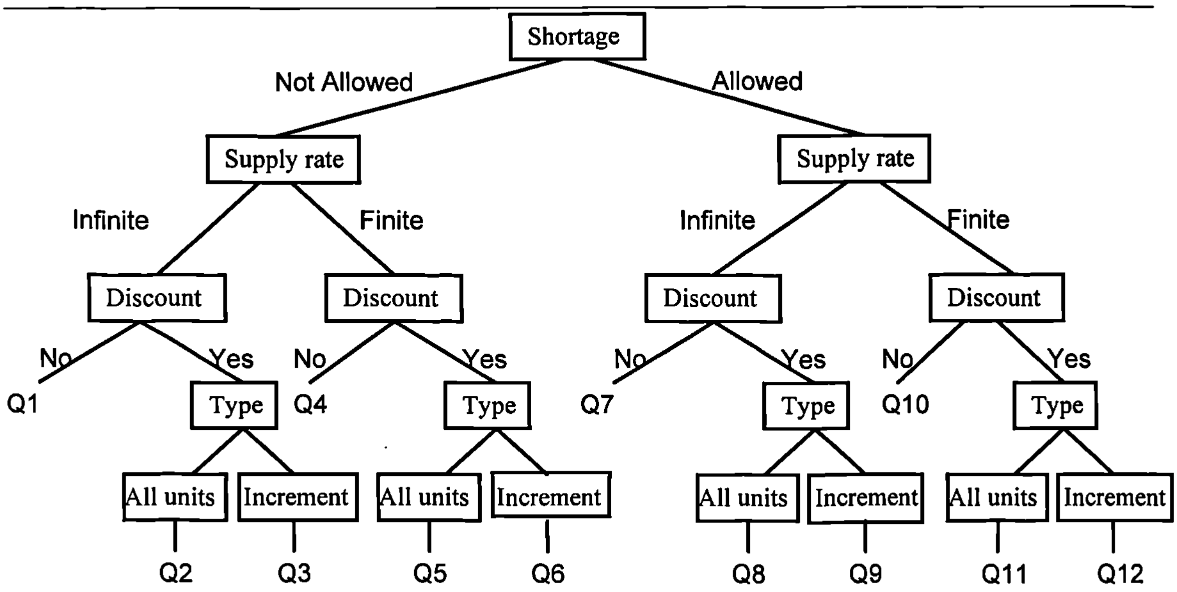


Figure 5.1 Model Tree of Constant Demand Inventories

Fig. 5.1 illustrates the possible types of the constant demand inventory models. The factors which are used in model selection for items with constant demand are whether shortage is allowed, the presence and types of unit price discount, and the supply rate. In the case of back ordering, there is a cost of shortage which is proportional to the quantity shortage.

In many practical situations if the order quantity (Q) exceeds a certain amount (break-point Q_b), the unit price (U_p) is discounted by a percentage (p). For example, for the case of a single break point, the total cost of the order, $C_p(Q)$, behaves as follows:

- All unit discount

$$\text{Batch purchasing cost} = C_p(Q) = \begin{cases} QU_p & 0 \le \mathrm{Q} < \mathrm{Q_b} \\ QU_p(1-p) & \mathrm{Q} \ge Q_b \end{cases} \tag{5-1}$$

- Incremental discount

$$\text{Batch purchasing cost} = C_p(Q) = \begin{cases} QU_p & 0 \le Q < Q_b \\ QU_p + (Q-Q_b)U_p(1\text{-p}) & \mathrm{Q} \ge Q_b \end{cases}. \tag{5-2}$$

If the holding cost is assumed to be proportional to the unit price of the item, the variable unit price will result in the holding cost having a different value when orders are made within different quantity breakpoints. The methods of estimating holding costs for quantity discounts are discussed in addition to different inventory models in the following section.

The supply rate describes how replenishment quantity arrives. The whole order may be received at the same time (infinite supply rate), or instead, it may become available at a rate of R_s (finite supply rate), e.g. if the ordered product is manufactured at a certain rate.

Models Q1, Q2, Q3, Q4, and Q7 of Fig. 5.1 which include only one factor (such as a planned shortage, discount, etc.) have simple formulas available in the literature. However, the other models, including Q5, Q6, Q8, Q9, Q10, Q11, and Q12, for situations with constant supply rate or with combined shortage and discount models are not available in the literature. Therefore, the procedure for calculating the optimal order quantity for these situations have been developed in this study. In the description of the following section, the development of model Q5, Q8, and Q11 are detailed in the main text because of their simple structure. For other models (Q6, Q9, Q10, Q12), only the results have been given directly with the derivation of these models detailed in Appendix 2.

5.2.1 Models for Constant Demand Inventories

For the inventory items with constant demand and constant lead time, the classical economic order quantity (EOQ) model and its modified and extended forms are employed. The following common notation is used in presenting the various models. Other notation specific to a particular situation is introduced during the discussion.

B = Optimal amount of shortage (in units);

$B^{(i)}$= Optimal amount of shortage at break point i (in units);

C_h = Holding cost (in pounds);

C_p= Ordering cost (in pounds);

C_s = Shortage cost (in pounds);

$C_h^{(i)}$= Holding cost of different discount break points (in pounds);

$C(q_i)$= Cost of purchasing an item for q_i units (and $C(q_0) = 0$) (in pounds);

D = Annual demand (in units);

$i = 0,1, 2, ..., m;$

I = The ratio of annual holding cost to purchasing price (%);

m = Number of discount break points;

q_i = Quantity of different discount break points ($q_0 = 0$) (in units);

$Q^{(i)}$= Order quantity according to break point i (in units);

R_s= Supply rate (units/unit time);

U_p= Unit price (in pounds);

$U_p^{(i)}$= Prices of different discount break points (in pounds);

$U_p^{(0)} = U_p$.

The following is a summary of the models and an outline of procedures used with constant demand inventory items.

1. EOQ model with infinite supply rate, no discount, and no shortage allowed [36]

a) Order quantity $$Q1 = \sqrt{\frac{2DC_p}{C_h}} \tag{5-3}$$

b) Total Cost $$TC(Q1) = \sqrt{2DC_pC_h} + DU_p \tag{5-4}$$

2. EOQ model with all units discount, infinite supply rate, and no shortage allowed [36]

Step 1

Compute an order quantity $Q^{(i)}$ for the unit price associated with each type of discount using

$$Q^{(i)} = \sqrt{\frac{2DC_p}{C_h^{(i)}}}; \tag{5-5}$$

where

$$C_h^{(i)} = I \cdot U_p^{(i)} \tag{5-6}$$

Step 2

For these order quantities which are too small to qualify for the assumed discount price, increase the order quantity to the nearest discount quantity that will allow the item to be purchased at the assumed price;

Step 3

Calculate the total cost for each of the order quantities

$$TC(Q^{(i)}) = \frac{Q}{2}C_h + \frac{D}{Q}C_p + DU_p \tag{5-7}$$

and then select the $Q^{(i)}$ with the minimum cost as the optimal order quantity Q2.

3. EOQ model with increment discount, infinite supply rate, and no shortage allowed [130]

Step 1

Compute an order quantity for every discount break point using

$$Q^{(i)} = \sqrt{2D(C_p + C(q_i) - q_i U_p^{(i)}) / C_h^{(i)}}, \tag{5-8}$$

where

$$C_h^{(i)} = I \cdot U_p^{(i)} \tag{5-9}$$

Step 2

Calculate the total cost for each order quantity which satisfies the condition $q_i < Q^{(i)} < q_{i+1}$ using

$$TC(Q^{(i)}) = DU_p{}^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - q_i U_p^{(i)}) + \\ + \frac{I}{2}(C(q_i) + Q^{(i)}U_p^{(i)} - q_i U_p^{(i)}) \tag{5-10}$$

Step 3

The $Q^{(i)}$ corresponding to the smallest $TC(Q^{(i)})$ is the optimal order quantity Q3.

4. EOQ model with finite supply rate, no discount; and no shortage allowed [1]

a) Order quantity $Q4 = \sqrt{\dfrac{2DC_p}{C_h(1-D/R_s)}}$ (5-11)

b) Total cost $TC(Q4) = DU_p + \sqrt{2DC_pC_h(1-D/R_s)}$ (5-12)

5. EOQ model with all units discount, finite supply rate, and no shortage allowed

Step 1

Compute each order quantity $Q^{(i)}$ for the unit price associated with each type of discount using

$$Q4 = \sqrt{\frac{2DC_p}{C_h^{(i)}(1-D/R_s)}}; \tag{5-13}$$

where

$$C_h^{(i)} = I \cdot U_p^{(i)} \tag{5-14}$$

Step 2

For these order quantities which are too small to qualify for the assumed discount price, increase the order quantity to the nearest discount quantity that will allow the item to be purchased at the assumed discount break price;

Step 3

Calculate the total cost for each of the order quantities using

$$TC(Q^{(i)}) = DU_p^{(i)} + \frac{Q(1-D/R_s)}{2}C_h^{(i)} + \frac{D}{Q}C_p \tag{5-15}$$

and then select the $Q^{(i)}$ with minimum cost as the optimal order quantity Q5.

6. EOQ model with increment discount, finite supply rate, and no shortage allowed (see Appendix 2)

Step 1

Compute an order quantity for every discount break point using

$$Q^{(i)} = \sqrt{2D(C_p + C(q_i) - q_iU_p^{(i)})/(C_h^{(i)}(1-D/R_s))}\ , \tag{5-16}$$

where

$$C_h^{(i)} = I \cdot U_p^{(i)} \tag{5-17}$$

Step 2

Calculate the total cost for each order quantity which satisfies the condition $q_i < Q^{(i)} < q_{i+1}$ using

$$\begin{aligned} TC(Q^{(i)}) = DU_p{}^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - q_iU_p^{(i)}) + \\ + \frac{I}{2}(C(q_i) + Q^{(i)}U_p^{(i)} - q_iU_p^{(i)})(1-D/R_s) \end{aligned} \tag{5-18}$$

Step 3

The $Q^{(i)}$ corresponding to the smallest $TC(Q^{(i)})$ is the optimal Q6.

7. EOQ model with shortage allowed, infinite supply rate, and no discount [1]

a) Order quantity $Q7 = \sqrt{\dfrac{2DC_p(C_s + C_h)}{C_h C_s}}$ (5-19)

b) Total cost $TC(Q7) = DU_p + \sqrt{\dfrac{2DC_p C_h C_s}{C_h + C_s}}$ (5-20)

c) Optimal maximum shortage $B = \sqrt{\dfrac{2DC_p C_h}{C_s(C_h + C_s)}}$ (5-21)

8. EOQ model with all units discount, infinite supply rate, and shortage allowed

Step 1

Compute each order quantity $Q^{(i)}$ for the unit price associated with each discount break point using

$$Q^{(i)} = \sqrt{\frac{2DC_p(C_s + C_h^{(i)})}{C_h^{(i)} C_s}}; \tag{5-22}$$

and each optimal maximum shortage using

$$B = \sqrt{\frac{2DC_p C_h^{(i)}}{C_s(C_h^{(i)} + C_s)}} \tag{5-23}$$

where

$$C_h^{(i)} = I \cdot U_p^{(i)}. \tag{5-24}$$

Step 2

For these order quantities which are too small to qualify for the assumed discount price, increase the order quantity to the nearest discount quantity that will allow the item to be purchased at the assumed discount break price;

Step 3

Calculate the total cost for each of the order quantity and holding cost using

$$TC(Q^{(i)}) = DU_p^{(i)} + \frac{(Q-B)^2}{2Q}C_h^{(i)} + \frac{D}{Q}C_p + \frac{B^2}{2Q}C_s \tag{5-25}$$

and then select the $Q^{(i)}$ with minimum cost as the optimal order quantity Q8.

9. EOQ model with increment discount, infinite supply rate, and shortage allowed (see Appendix 2)

Step 1

Compute an order quantity for every discount break point using

$$Q^{(i)} = \sqrt{\frac{2D(C_p + C(q_i) - q_iU_p^{(i)})(C_h^{(i)} + C_s)}{C_h^{(i)}C_s}}\ , \tag{5-26}$$

where $C_h^{(i)}$ is estimated by

$$C_h^{(i)} = I \cdot (\sum_{j=0}^{i} U_p^{(j)}) / (i+1) \tag{5-27}$$

Step 2

Calculate the total cost for those order quantities which hold the condition $q_i < Q^{(i)} < q_{i+1}$ using

$$\begin{aligned} TC(Q^{(i)}, B^{(i)}) = DU_p{}^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - q_iU_p^{(i)}) + \\ + \frac{C_h^{(i)}}{2}(Q^{(i)} - B^{(i)})^2 + \frac{C_sB^{(i)2}}{2Q^{(i)}} \end{aligned} \tag{5-28}$$

where

$$B^{(i)} = \frac{C_h^{(i)}}{C_h^{(i)} + C_s}Q^{(i)} \tag{5-29}$$

Step 3

The $Q^{(i)}$ corresponding to the smallest $TC(Q^{(i)}, B^{(i)})$is the optimal Q9.

10. EOQ model with finite supply rate, shortage allowed, and no discount (see Appendix 2)

a) Order quantity $Q10 = Q = \sqrt{\frac{2DC_p}{C_h} \cdot \frac{R_s}{R_s - D} \cdot \frac{C_s + C_h}{C_s}}$ (5-30)

b) Total cost

$$TC(Q,B) = DU_p + \frac{D}{Q}C_p + \frac{C_s B^2 R_s}{2Q(R_s - D)} + \\ + \frac{C_h}{2}(Q(1-\frac{D}{R_s}) - B)(1 - \frac{B}{Q} - \frac{BD}{Q(R_s - D)}) \tag{5-31}$$

where

$$B = \frac{C_h(R_s - D)}{R_s(C_s + C_h)}Q \tag{5-32}$$

11. EOQ model with all units discount, finite supply rate, and shortage allowed

Step 1

Compute each order quantity $Q^{(i)}$ for the unit price associated with each type of discount (see Appendix 2) using

$$Q^{(i)} = \sqrt{\frac{2DC_p}{C_h^{(i)}} \cdot \frac{R_s}{R_s - D} \cdot \frac{C_s + C_h^{(i)}}{C_s}}; \tag{5-33}$$

where

$$C_h^{(i)} = I \cdot U_p^{(i)} \tag{5-34}$$

Step 2

For these order quantities which are too small to qualify for the assumed discount price, increase the order quantity to the nearest discount quantity that will allow the item to be purchased at the assumed discount break price;

Step 3

Calculate the total cost for each of the order quantity using

$$TC(Q^{(i)}, B^{(i)}) = DU_p^{(i)} + \frac{D}{Q^{(i)}}C_p + \frac{C_s B^{(i)2} R_s}{2Q^{(i)}(R_s - D)} + \\ + \frac{C_h^{(i)}}{2}(Q^{(i)}(1-\frac{D}{R_s}) - B^{(i)})(1 - \frac{B^{(i)}}{Q^{(i)}} - \frac{B^{(i)}D}{Q^{(i)}(R_s - D)}) \tag{5-35}$$

where

$$B^{(i)} = \frac{C_h^{(i)}(R_s - D)}{R_s(C_s + C_h^{(i)})} Q^{(i)} \tag{5-36}$$

and then select the $Q^{(i)}$ with minimum cost as the optimal order quantity Q11.

12. EOQ model with increment discount, finite supply rate, and shortage allowed (see Appendix 2)

Step 1

Compute an order quantity for every discount break point using

$$Q^{(i)} = \sqrt{\frac{2D(C_p + C(q_i) - q_i U_p^{(i)})}{C_h^{(i)}} \cdot \frac{R_s}{R_s - D} \cdot \frac{C_s + C_h^{(i)}}{C_s}} \tag{5-37}$$

where $C_h^{(i)}$ is estimated by

$$C_h^{(i)} = I \cdot (\sum_{j=0}^{i} U_p^{(j)}) / (i+1) \tag{5-38}$$

Step 2

Calculate the total cost for those order quantities which holds the condition $q_i < Q^{(i)} < q_{i+1}$ using

$$\begin{aligned} TC(Q^{(i)}, B^{(i)}) = DU_p^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - q_i U_p^{(i)}) + \frac{C_s B^{(i)2} R_s}{2Q^{(i)}(R_s - D)} + \\ + \frac{C_h^{(i)}}{2}(Q^{(i)}(1 - \frac{D}{R_s}) - B^{(i)})(1 - \frac{B^{(i)}}{Q^{(i)}} - \frac{B^{(i)} D}{Q^{(i)}(R_s - D)}) \end{aligned} \tag{5-39}$$

where

$$B^{(i)} = \frac{C_h^{(i)}(R_s - D)}{R_s(C_s + C_h^{(i)})} Q^{(i)} \tag{5-40}$$

Step 3

The $Q^{(i)}$ corresponding to the smallest $TC(Q^{(i)}, B^{(i)})$ is the optimal order quantity Q12.

5.2.2 Ordering Decision

A new order is made for any item when its stocking level is equal or less than its reorder point which coincides with the amount of demand during the lead time. For the constant inventory problems, all the factors that affect the value of order-point, such as demand and lead time are constant and known beforehand. Therefore it is better and possible not to have any stock on hand when a new order arrives. The methods for calculating the order-point depend on the length of the lead time:

- If the lead time (L) is less than the time between two successive orders (T),

$$\textit{Reorder-point (s)} = \textit{Constant demand per unit time} \times \textit{Lead time (L)} - B. \quad (5\text{-}41)$$

In the case with zero lead time ($L = 0$),

$$\textit{Reorder-point (s)} = B. \quad (5\text{-}42)$$

- If the lead time (L) is greater than the time between two successive orders (T),

$$\textit{Reorder-point (s)} = \textit{Constant demand per unit time} \times \textit{Lead time (L)} - B. \quad (5\text{-}43)$$

In this case, an order decision is made based on gross inventory (net inventory + on-order), i.e.

$$\textit{Net-inventory} + \textit{On-order} \leq \textit{Reorder-point.}$$

5.3 Models for Inventory Items with Probabilistic Demand

The probabilistic inventory problems (see Fig 5.2) include the cases of constant demand with probabilistic lead time, probabilistic demand with constant lead time, and probabilistic demand with probabilistic lead time.

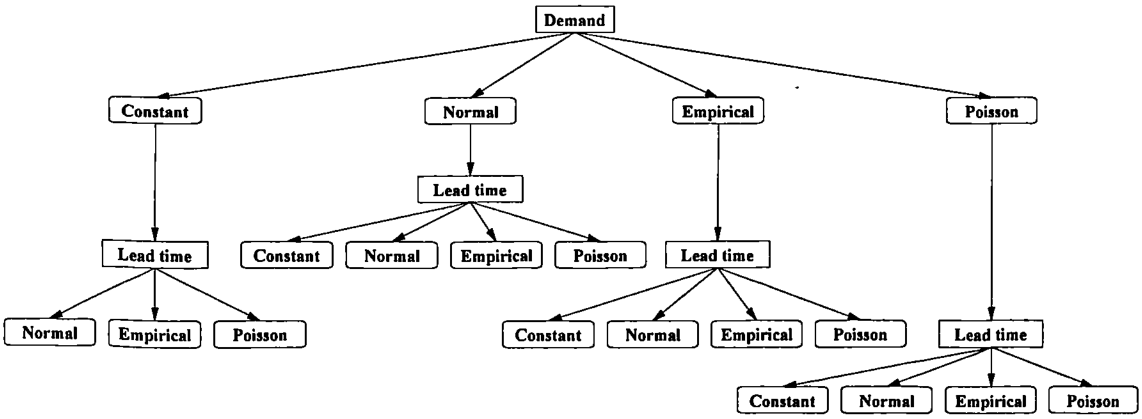


Figure 5.2 Model Tree for Probabilistic Inventory Problems

5.3.1 Comparison between Different Inventory Policies

The inventory policies for items with probabilistic demand are summarised in Table 5.1 [131].

The periodic-review, order-up-to-level (*T, S*) system is also known as a replenishment cycle system. The control procedure is that every *T* units of time enough quantity is ordered to raise the stock level to the maximum stock level *S*.

Table 5.1 Inventory Policies for Probabilistic Demand

Policy	Reorder control parameter	Reorder quantity
(*T, S*) system	*T*	Restore to *S*
(*s, S*) system	*s*	Restore to *S*
(*s, Q*) system	*s*	Fixed *Q*
(*T, s, S*) system	*T, s*	Restore to *S* or no reorder

s=reorder point; Q=order quantity; S=maximum inventory level T=review interval.

The order-point, order-up-to-level (*s, S*) system is a continuous review policy and a new order is made whenever the stock level drops to the reorder point *s* or lower. However, in this system a variable order quantity is used to raise the stock level to the maximum stock level *S* whenever an order is placed.

The reorder-point, order-quantity (*s, Q*) system is also a continuous review policy. A fixed quantity *Q* is ordered whenever the stock level drops to the reorder point *s* or lower. A new order may be triggered immediately after every withdraw from the storeroom.

The periodic review, reorder-point, order-up-to-level (*T, s, S*) system is a combination of (*s, S*) and (*T, S*) systems. The idea is that every *T* units of time the stock level is checked. If it is at or below the reorder point *s*, an order is made to raise the stock level to the maximum stock level *S*. If the stock level is above reorder point *s*, nothing is done until at least the next review.

The (*T*, *S*) and (*T*, *s*, *S*) policies are periodic review policies. The main disadvantage of these polices is that, to provide the same level of customer service, it requires more safety stock, hence higher holding costs than those for the continuous review policies. In addition, a periodic review may cause an increase of the risk of stockout. By means of a computerised inventory control system, the inventory status can be easily monitored. Therefore, the two continuous review policies, (*s*, *Q*) and (*s*, *S*) are preferred. One disadvantage of (*s*, *Q*) system is that if the transaction that triggers the replenishment is large enough, then a replenishment of size *Q* will not even raise the inventory position above the reorder point. Thus the continuous review, order-point, order-up-to-level (*s*, *S*) policy is adopted in this study.

The major advantage of (*s*, *S*) system is that it has been shown to have a total cost no larger than those of the (*s*, *Q*) systems [36]. Moreover, (*s*, *S*) system is frequently used in practice [132]. A possible disadvantage of the (*s*, *S*) system is the danger of errors in requisition caused by the variable order quantity [36]. But this should not represent any real difficulty with a modern computerised system.

5.3.2 Calculating the Values of Parameters Used in (*s*, *S*) Policy

When demand (or lead time) is probabilistic, there is a definite chance of not being able to satisfy some demand. To avoid stockout a certain amount of safety stock is carried. The methods of estimating the needed safety stock depend on the managerial attitudes toward what factors (such as costing shortage, common factor, service level, and aggregate consideration) are taken into account [1]. There are no hard and fast rules for selecting the appropriate approach. In this study the service level (S_L), which is a specified probability of no stockout per replenishment cycle, is adopted to develop the safety stock models and the basic decision criterion used with the probabilistic inventory models is to satisfy the specified service level [3]. To explain the theoretical derivation of the decision rules the following notations are introduced:

$\overline{D}_L$ = Average demand over lead time (in units);

D_L = Demand over lead time;

σ_d = Standard deviation of demand per unit time over lead time (in units/unit time);

σ_D = Standard deviation of demand over lead time;

σ_L = Standard deviation of lead time (in days, weeks, etc.);

d_L = Constant demand per unit time over lead time (in units/unit time);

$\overline{d_L}$= Average demand per unit time over lead time (in units/unit time);

L= Constant lead time (in days, weeks, etc.);

$\bar{L}$= Average lead time (in days, weeks, etc.);

Q_s= Safety stock (in units);

s = Reorder-point (in units);

S = Maximum inventory level (in units);

S_L= Service level (%);

Z = Number of standard deviations of demand corresponding to the service level (safety factor);

$P_u \geq (z)$= Probability of a variable takes on value of Z or large.

To decide the safety stock for the item with normal distribution, suppose management has specified that the probability of no stockout should not be lower than S_L, then we have

$$\text{Prob}\{\text{stockout over delivery lead time}\}= P_u \geq (z).$$

Because the desired service level is S_L, then we must have

$$\text{Prob}\{\text{stockout over delivery lead time}\}=1\text{-}S_L,$$

Then we have to select the safety factor Z to satisfy

$$P_u \geq (z) = 1 - S_L.$$

which can be expressed as a product of safety factor Z and demand standard deviation during lead time σ_D , i.e. [36]

$$\text{Safety stock } (Q_s) = Z \times \sigma_D.$$

If the demand in the replenishment lead time has a probability density function $F_d(d_L)$, the net stock just before the replenishment arrives is

$$Q_s = \int_0^{\infty} (s - D_L) f_d(D_L) dD_L, \text{ i.e. } Q_s = \text{s} + \overline{D}_L,$$

That means the reorder point is decided by the safety stock and average demand during the lead time, i.e.

Reorder-point (s) = Average demand during lead time ($\overline{D}_L$) + Safety stock (Q_S).

The estimation of reorder-point of items with Poisson or empirical distributions are discussed in the following section.

5.3.2.1 Calculating the Reorder-point

In the continuous *(s, S)* inventory system, a new order is triggered when the current inventory status is equal or less than the reorder-point *s*. Peterson and Silver [36] have suggested that the reorder-point *s* in a continuous (*s*, *S*) system can be estimated by the same method used for an (*s*, *Q*) system. The details of how to calculate the reorder point for the different probabilistic inventory situations are described below.

a. Probabilistic Demand and Constant Lead Time

- If the demand follows a normal distribution, the reorder point is given by

$$s = \bar{d}_L \cdot L + Z \cdot \sigma_d \cdot \sqrt{L}. \qquad (5\text{-}44)$$

- If the demand follows a Poisson distribution, the reorder point is the maximum value of m that satisfies

$$\sum_{n=0}^{m}\frac{e^{-\lambda L/t}(\lambda L/t)^{n}}{n!}\le S_L. \tag{5-45}$$

- If the demand follows an empirical distribution, the reorder point is the maximum value of m that satisfies

$$\sum_{n=0}^{m} f_n^{(D)}\le S_L \tag{5-46}$$

where $f_n^{(D)}$ = the frequency of the category n of the demand.

b. Constant Demand and Probabilistic Lead Time

- If the lead time follows a normal distribution, the reorder point is given by

$$s=d_L\cdot\bar{L}+Z\cdot d_L\cdot\sigma_L. \tag{5-47}$$

- If the lead time follows a Poisson distribution, the reorder point is

$$s=d_L\cdot L_m. \tag{5-48}$$

where L_m is the maximum value of m that satisfies

$$\sum_{n=0}^{m}\frac{e^{-\lambda}(\lambda)^{n}}{n!}\le S_L. \tag{5-49}$$

- If the lead time follows an empirical distribution, the reorder point is

$$s=d_L\cdot L_f. \tag{5-50}$$

where the L_f is the maximum value of m that satisfies

$$\sum_{n=0}^{m} f_n^{(L)}\le S_L \tag{5-51}$$

and $f_n^{(L)}$ = the frequency of the category n of the lead time.

c. Probabilistic Demand and Probabilistic Lead Time

If both the demand and lead time have Normal distributions, the reorder point is given by [36]

$$s=\bar{d}_L\,\bar{L}+Z\cdot\sqrt{\sigma_d^2\,\bar{L}+\sigma_L^2\,\bar{d}_L^{2}}\,. \tag{5-52}$$

5.3.2.2 Calculating Standard Deviation of Demand During Lead Time

Sometimes the standard deviation of demand and average demand are known, but over a time interval t which is different from the lead time. For example, records may report only a monthly demand while the lead time may be measured in days. In this situation the following conversions can be used [133]:

$$\sigma_D = \sigma_t \sqrt{\frac{L}{t}} \tag{5-53}$$

where σ_t is the standard deviation of demand per time interval t.

When using this formula, both L and t are expressed in the same time unit. After applying the conversion formula, the resulting σ_D can be used to find the safety stock (Qs).

5.3.2.3 Calculating the Current Inventory Status

The current inventory status is defined by the relation

Current inventory status (S_c) = (On-hand) + (On-order) - (backorders).

A new order is triggered when the current inventory status is less than the reorder point.

5.3.2.4 Calculating the Maximum Inventory Level

The maximum inventory level for an item is given by

Maximum inventory level (S) = Reorder point (s) + Economic order quantity ($\tilde{Q}$)

The EOQ is calculated by [36]

$$\tilde{Q} = \sqrt{\frac{2DC_p}{C_h}}.$$

5.3.2.5 Calculating the Order Quantity

The order quantity (*Q*) is the difference between the maximum inventory level and the current inventory status, i.e.

Order Quantity (Q) = Maximum inventory level (S) - Current inventory status (S_c).

5.4 Model for Items with Low Demand

According to the features of low demand items described in Section 4.2, the (*s, s+1*) policy is adopted to cope with this kind of inventory items. This policy is a special case of (*s, S*) policy with *S = s+1 or Q = 1*. Therefore, the only decision parameter needs to be estimated is s which is calculated according the patterns of lead time.

1. If the lead time is constant, the reorder point is

$$s = \overline{d_L} \cdot L \tag{5-54}$$

2. If the lead time is probabilistic, the reorder point is calculated by the same methods used in calculating the reorder point for items with constant or probabilistic demand and probabilistic lead time (see Section 5.3.2.1).

In the above discussion, the continuous review (*s, S*) policy is adopted to deal with inventory items with probabilistic demand and low demand. In fact, the periodic review (*s, S*) policy is also in common use particularly in companies not utilising computer based inventory management systems. However, there is so far little guidance available to assist organisations in selecting the best periodic policy to adopt from among the many heuristics for real practical situations although the (*s, S*) policy has been claimed theoretically to be the best for the management of low and intermittent demand items [134].

For companies which adopted a periodic review inventory management system, the values of parameters such as review period (*T*), reorder point (*s*), and maximum inventory level (*S*) should be estimated in different ways [36]. Sani and Kingsman's [134] have concluded that a company can choose one of the existing model for its inventory management because none of them is significantly better than others at all time by comparing ten periodic inventory policies in according to five demand forecasting methods.

5.5 Models for Inventory Items with Linear Demand

For the cases of increasing linear demand patterns, the instantaneous demand rate at time *t*, *D(t)*, is assumed to be large enough to be reasonably approximated by the function

$$D(t) = a + bt + \varepsilon_t$$

where

ε_t = demand fluctuation;

a = The demand rate at time 0;

b = The trend value.

Similarly, the decreasing linear demand patterns are presented as

$$D(t) = a - bt + \varepsilon_t$$

In the following discussion, *H* is introduced to represent the length of the planning horizon and the lead time is assumed to be constant.

5.5.1 Review of Existing Models

The earliest model to deal with inventory problems with linear trend demand was presented by Diegel [135] for the particular case where the replenishment rate is constant. The difference S_i between replenishment and demand accumulating over an interval $t_i = T_i - T_{i-1}$ is defined by

$$S_i = \int_{T_{i-1}}^{T_i} [Q_{i-1} + (t - T_i)(a + 0.5b(T_i + T_{i-1})) - at - bt^2 / 2]dt$$

where Q_{j-1} is the sum of previous *j-1* orders.

The minimisation of the total accumulated discrepancies $Z_j = \sum_{i=1}^{j} S_i$ over the time of a sequence of *j* replenishments yields;

$$t_i = t_{i-1} \tag{5-55}$$

That is, all replenishments should take place at an equal interval of time; hence, $t_i = H / n$ that leads to the optimal replenishment number

$$n = \sqrt[3]{|bc_h H^3 / 6c_p|} \tag{5-56}$$

and the optimal reorder quantity

$$Q_i = aH / n + (2i - 1)bH^2 / 2n^2, \quad n = 1, 2, \ldots, n. \tag{5-57}$$

Obviously, this model does not minimise the total relevant inventory costs but only reduces the accumulated difference between the replenishment and demand over an interval.

The research for the general case of a positive linear trend in demand was presented by Donaldson [136]. First, he established that the order quantity at a replenishment point should be the product of the current instantaneous damned rate and the elapsed time since the last replenishment, i.e.

$$\int_{t_i}^{t_{i+1}} f(u)du = (t_i - t_{i-1}) f(t_i) \qquad i = 1, 2, \ldots\ldots n\text{-}1. \tag{5-58}$$

Donaldson [136] then used this property to determine the optimal number and timing of replenishments. The optimal value of n occurs when $c_p / H^3bc_h = g(n+1)$ or when *g(n+1)*, for the first time, becomes less than c_p / H^3bc_h, where *g(n+1)* is defined as

$$g(n) = G(n) - G(n-1), \qquad \mathrm{n} \geq 2 \tag{5-59}$$

and

$$G(n) = \begin{cases} (1/\mathrm{g}_\mathrm{n}^3 \sum_{\mathrm{i}=1}^{\mathrm{n}-1} \mathrm{g}_\mathrm{n}^3[1-(1/\mathrm{Z}_\mathrm{i}), & \mathrm{n} \geq 2 \\ 0, & \mathrm{n} = 1 \end{cases} \tag{5-60}$$

The optimal replenishment times, $t_1, t_2, \ldots, t_n$, are then calculated as follows:

$$t_1 = H / g_n \tag{5-61}$$

and in general

$$t_i = g_i / g_n . H, \qquad t_0 = 0, \tag{5-62}$$

where

$$g_i = \prod_{r=2}^{i} Z_r \ , \ \mathrm{g}_1 = 1, \ Z_{i+1} = \sqrt{3-(2/Z_i)}, \ Z_1 = \sqrt{3}, \text{ and } i=1,2, \ldots, n. \tag{5-63}$$

The optimal solution is straightforward, but the computational effort of this model is not simple, particularly when there is a positive demand at the start of the period. Moreover, if the demand does not cease at the end of the planning horizon, the model can only give a sub-optimal solution, and the situation would be reviewed and the new reorder points recalculated.

Louis and Brosseau[137] also presented a very intuitive replenishment policy for the linearly decreasing demand inventory problem. The solution for any given problem is obtained by dividing the total relevant cost (*TRC*) of Donaldson's model

$$TRC = nc_p + \frac{H^3bc_h}{6} - \frac{bc_h}{2} \sum_{i=1}^{n-1} t_i((H-t_i)^2 - (H-t_{i+1})^2) \tag{5-64}$$

by H^3bc_h to normalise the total relevant cost (*NTRC*) over *H*, namely

$$NTRC\,(n, M) = nM + \frac{1}{6} - \frac{1}{2H^3} \sum_{i=1}^{n-1} t_i((H-t_i)^2 - (H-t_{i+1})^2), \tag{5-65}$$

where

$$M = c_p b^2 / c_h a^3, \qquad H = a / b\ . \tag{5-66}$$

Given different numbers of replenishments (*n=1, 2, ...*), equation (5-65) can be represented by the single parameter *M.* e.g.

$$\begin{aligned} NTRC(1, M) &= M + \frac{1}{6} \\ NTRC(2, M) &= 2M + \frac{5}{54} \\ NTRC(3, M) &= 3M + \frac{205}{3174} \\ &\vdots \\ &\vdots \end{aligned}$$

The ranges of *M* for which a particular replenishment number *n* is optimal are obtained by solving the entire set of indifference conditions of normalised total relevant costs between *n-1* and *n,* i.e.

$$\begin{aligned} NTRC(1, M) &= NTRC(2, M) \\ NTRC(2, M) &= NTRC(3, M) \\ &\vdots \end{aligned}$$

To avoid the laborious calculation, tabular results for the analytical solution have been derived to allow the practitioner to read off the number of orders to replace, and when they should be replaced with a very little computational effort. But the problem with this approach is that it can only cope with inventory items for which demand is declining toward zero over a certain period of time. Moreover, it is impossible to implement the model into a computer system because there is no common formula to express the normalised total relevant costs for any replenishment number *n*.

The prime purpose of studying theoretical models is to improve decision making in real-life situations. Thus it should be recognised that the analytical solution is just a device to aid the mathematical solution and is not an essential feature of real life situations. The main value of these formulations is that they have led to the development of other simple but more practical heuristics.

To let the replenishments take place at equal interval T, Phelps [138] proposed a model which, unlike other models, applies to both positive and negative trends. The total cost is assumed to be a continuous function of the number of replenishments n,

$$TC(n) = nc_p + c_h \sum_{i=0}^{n-1} \int_{iT}^{(i+1)T} (t - iT)D(t)dt \tag{5-67}$$

The optimal value of n which minimises the cost is given by

$$n^3 = (H^2(2a + Hb)c_h / 4c_p)n + H^3bc_h / 6c_p. \tag{5-68}$$

For the positive trend, $b > 0$, the equation can be solved using the form

$$n_{k+1} = c_h\sqrt{H^3b} / 6c_pn_k + H^2(2a + Hb)c_h / 4c_p. \tag{5-69}$$

For the negative trend, $b < 0$, if and only if $c_p / c_h - H^2(6a + 5Hb) / 12 < 0$, there is a solution.

Ritchie [139] proposed a solution which has the simplicity of the EOQ formula by increasing the replenishment number n (or unbound the horizon H). Based on Donaldson's work [136] Ritchie proposed a model which simplifies the first replenishment t_1 as

$$t_1 = \sqrt{c_p / 0.34bc_h}. \tag{5-70}$$

Subsequent replenishment times are calculated using Donaldson's model.

Alternatively, replenishment quantity at t_i is evaluated by

$$Q_{t_i} = bt_i(t_i - t_{i-1}). \tag{5-71}$$

Later Ritchie and Tsado [140] extended the previous work to the particular case of $a=0$. That gives

$$Q_{t_i} = G_i \cdot EOQ, \tag{5-72}$$

where

$$G_i = \frac{(g_i - g_{i-1})g_i^{1/2}}{\sqrt{0.86}}, \quad g_i = \prod_{r=2}^{i} Z_r \ , g_1 = 1, \ Z_{i+1} = \sqrt{3 - (2/Z_i)}. \text{ and } Z_1 = \sqrt{3}.$$

There is nothing new in this model as it repeats an ad hoc of Donaldson's model when initial demand is zero. The calculation procedure is as complicated as Donaldson's model.

By simply minimising the total relevant cost per unit time(*TRCUT*) over the first duration of replenishment *T*, Silver [141] developed a heuristic for the inventory items with a positive trend in demand. To select the value of *T* minimise

$$TCUT(T) = [c_p + c_h \int_0^T tD(t)dt] / T, \tag{5-73}$$

which leads to

$$T^2[(a/2) + (2/3)bT = c_p / c_h.. \tag{5-74}$$

For the case of $a = 0$, a closed solution for *T* is

$$T = \sqrt[3]{(3c_p / 2bc_h)}. \tag{5-75}$$

otherwise, the replenishment intervals are given by

$$T^{(i+1)} = [2c_p / (a + 4/3)bc_h T^{(i)}]^{1/2}, \tag{5-76}$$

where $T^{(i)}$ is the value of *T* obtained on the *i*th iteration, and selecting $T^{(0)} = 0$.

Instead of minimising the per unit time total relevant cost over an interval, Tsado [142] developed a model which minimises the total relevant cost per item over the interval, i.e. to minimise

$$TCU(T) = [c_p + c_h \int_k^{k+T} tD(t)dt] / [\int_k^{k+T} D(t)dt]. \tag{5-77}$$

This leads to

$$T^2[(a'/2) + (1/6)bT] = c_p / c_h. \tag{5-78}$$

For the case of positive trend, set $T^{(0)} = 0$. The replenishment points can be solved using

$$T^{(i+1)} = [2c_p / (a' + (1/3)bc_h T^{(i)}]^{1/2}. \tag{5-79}$$

In terms of the principle for the constant demand that the optimal replenishment quantity equalise the ordering and holding costs, i.e.

$$c_h \int_0^T tD(t)dt = c_p. \qquad (5\text{-}80)$$

Goyal and Gommes [143] derived a model that calculates the replenishment points with the iteration:

$$T^{(i+1)} = [2c_p / (a'(2/3)bc_h T^{(i)}]^{1/2}. \qquad (5\text{-}81)$$

The common form of some heuristics allows an eclectic model to be proposed. Therefore, Amrani and Rand [144] constructed a cubic equation by introducing the constant λ namely,

$$T^2[(a'/2) + \lambda bT] = c_p / c_h. \qquad (5\text{-}82)$$

This model will be analogous to Ritchie's model when $\lambda = 0.43$, to Silver's model when $\lambda = 2/3$, to Tsado's model when $\lambda = 1/6$ and to Goyal's model when $\lambda = 1/3$. The best decision is made by simply calculating the results for the four heuristics and then choosing the best one as the replenishment policy. While this model unifies the previous heuristics it heavily increases the burden of calculation.

From the above literature review, it appears that the published inventory models for items with linear trend in demand are generally focused on the numerical calculation for determining the replenishment points in time and the number of reorders within a given horizon. These publications which analyse the solutions for the problem from different angles lead to substantial benefits for practitioners and to possible research ideas. However, these models or algorithms also have serious limitations:

1. Only a highly qualified practitioner would be able to comprehend the analytical solutions of the model and the process of deriving the model.

2. Diegel's models [135] can only be used at the particular case with linearly increasing demand and constant replenishment rate that changes at every interval

according to the average demand during the interval. This may restrict the possibility of its practical application.

3. The method to decide the optimal replenishment number n suggested by Donaldson's model [136] is quite robust for the case of $a = 0$. For example, The model will give the same replenishment intervals for all problems if their parameters lie within the range of $0.014 < c_p / H^3bc_h \leq 0.024$ (see Table 5.2). Moreover, Donaldson's optimal solution is based on a estimated horizon H where the demand is supposed to cease. In practice this coincidence may not occur, in such cases the solution will be sub-optimal.

4. Unlike the analytical solutions all the heuristic models do not minimise the entire total costs over the horizon, but they only optimise the costs over each single interval in order to simplify the analysis and the calculations. They are neither as accurate as the analytical solutions nor as simple as Mitra's et al [145] model that will be selected for the system to deal with the linear demand inventory problems.

5. Ordering at different time points is difficult to administer in practice. Although Phelps's model [137] suggests ordering at constant intervals, but it involves solving a cubic equation which may be criticised for its complexity.

Table 5.2 Examples to Illustrate Indifference in Ordering Points of Donaldson's Model [136]

Example	Parameters					Results		
	c_p	c_h	a	b	H	c_p / H^3bc_h	No. of orders	Ordering Points
1	9	2	0	200	1	0.023	4	0, 0.34, 0.60, 0.81
2	9	2	0	300	1	0.015	4	0, 0.34, 0.60, 0.81
3	10	1	0	120	2	0.010	4	0, 0.34, 0.60, 0.81
4	20	1	0	120	2	0.024	4	0, 0.34, 0.60, 0.81
5	22	1	0	125	2	0.022	4	0, 0.34, 0.60, 0.81
6	25	1.5	0	125	2	0.017	4	0, 0.34, 0.60, 0.81

5.5.2 Analysis and Selection of a Model

From the practical point of view, a good inventory model should feature these advantages: 1) the analysis process of obtaining the model is straightforward and easy to understand; 2) the "how many and when to order" decisions made by the model do not include complicated algorithms; 3) the model has less restrictions and can cope with wide range of inventory problems; 4) the application of the model will lead to efficient inventory management and significant cost savings.

The procedure presented by Mitra's et al [145] modifies the EOQ model to accommodate the special cases of increasing and decreasing linear demand patterns. The first step of the method is to determine the overall demand over the horizon (*H*), i.e.

$$D = \int_0^H (a+bt)dt = aH + \frac{bH^2}{2}. \qquad (5\text{-}83)$$

Then the equivalent constant demand rate which is estimated from

$$D^* = \frac{1}{H}\int_0^H (a+bt)dt = a + bH/2, \qquad (5\text{-}84)$$

and the order interval is calculated by

$$T^* = \sqrt{2c_p / c_h D^*}. \qquad (5\text{-}85)$$

Thus the number of orders over *H* is

$$n = [D / D^* T^*]. \qquad (5\text{-}86)$$

that means *n* is the smallest integer greater than or equal to $D / D^* T^*$ and *D* is the overall demand during the horizon. The order quantity at each reorder point is the amount needed to satisfy the demand over the order interval, that is

$$\begin{aligned} Q_i &= \int_{(i-1)T^*}^{iT^*} D(t)Dt = aT^* + b(i - 0.5T^{*^2}) \quad \mathrm{i = 1, \ldots, n\text{-}1.} \\ &\vdots \\ &\vdots \\ Q_n &= D - \sum_{i=1}^{n-1} Q_i \end{aligned} \qquad (5\text{-}87)$$

As a result of its computational simplicity and without any heuristic adjustments. this method provides a practical alternative to the other approaches. Further, it has the advantages that: (1) the procedure is straightforward, thus it is easier for practitioners to understand and to implement this model in practice; (2) the operationally desirable feature that replenishments are made at equal intervals makes the model much easier to administrate; (3) it is equally applicable to the negative trend where analytical methods are not presently available; (4) this model can be smoothly transferred from linear demand to steady demand to deal with the cases that the demand may become steady after a period of increase (or decrease); (5) this model can be used to estimate the exact amount to cover the demand over a period to deal with the items with negative trends in demand which will not be stored after a certain period.

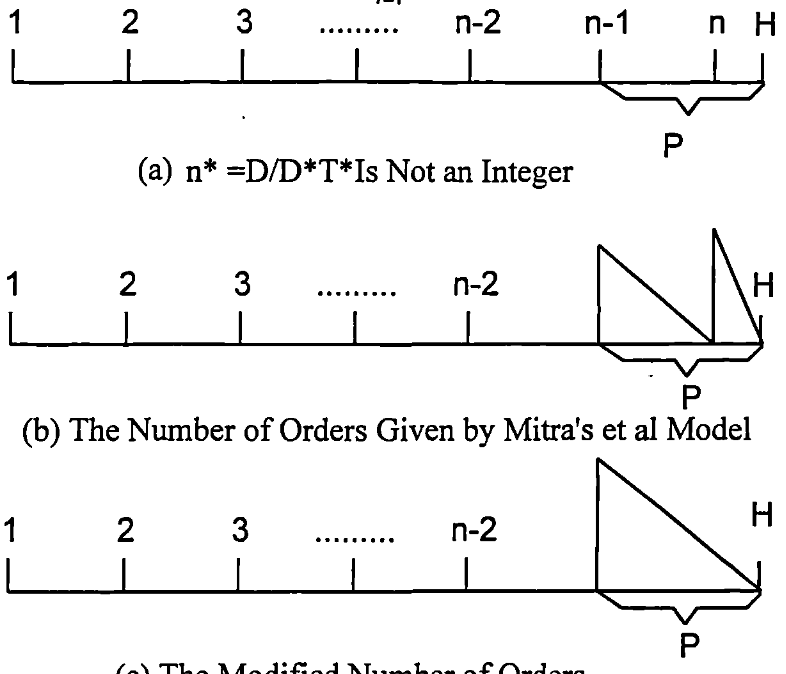
The model proposed by Mitra's et al [145] however has several shortcomings. First, the method can only work when the horizon is known in advance. This assumption is very reasonable because: (1) companies usually make their routine decisions in line with the time length of the financial year; (2) the trend may cease in the predictable future; (3) the items with decreasing demand would not be reordered after certain time; (4) some of the parameters, such as the holding cost, are calculated per unit time (financial year). Secondly. the use of this model may cause a slightly higher cost penalty than other analytic models, however a number of tests of this model have shown that the increase in cost over that of the optimal model is typically in the order of few percentage points (see Table 5.3). This cost increase is not significant compared with other models especially derived for this type of demand situation that had been investigated by Ritchie and Tsado [146]. Thirdly, the cost of this model will increase with the length of the time horizon. However, in practice any model for use over a long period would need to be updated at regular intervals to take account of the actual changes in demand, so this is unlikely to prove a practical disadvantage. Finally, the model gives equal replenishment intervals and thus must give excessive holding costs when demand is high (near the end of the horizon) and excessive ordering costs when demand is low (at the beginning of the horizon). This problem can be solved by adjusting the order quantities, i.e. to increase Q in the first half horizon and decrease Q in the last half horizon respectively.

The Mitra's et al model [145] suggests an extra order (see Equation 5-86) to cover the interval between the *n-1*th order and the end of the horizon H for the case that $n^* = D / D^*T^*$ is not an integer (Fig. 5.2). However, it may cost more if the extra holding cost of ordering a quantity at *n-1* period to cover the rest of the horizon is less than the ordering cost. In this case we should compare the total cost, R, to make only one order over the interval P the total cost, R', to make two orders at *n-1*th and *n*th period respectively to cover the interval P. if $R > R'$, an extra order should be placed at *n*th period. Otherwise, only *1* order should be made during the interval P. The total costs for these two reordering policies are:

$$R = c_h\left(\frac{TQ_{n-1}}{2H} + \frac{H-nT}{2H}\left(D-\sum_{i=1}^{n-1} Q_i\right)\right) + C_p \tag{5-88}$$

and

$$R' = \frac{C_h(H-(n-1)T)}{2H}\left(D-\sum_{i=1}^{n-2} Q_i\right) \tag{5-89}$$



(a) n* =D/D*T*Is Not an Integer

(b) The Number of Orders Given by Mitra's et al Model

(c) The Modified Number of Orders

Figure 5.3 The Explanation of Reorder Intervals

Table 5.3 Comparison of the Operating Cost of Donaldson's Model [136] and Mitra's et al Model [145]

Sample	Parameters					Total Cost		
No.	a	b	c_p	c_h	H	Mitra	Donaldson	difference %
1	6	1	9	0.1	11	51.23	51.35	-0.2
2	0	50	9	2	1	32.13	32.09	0.14
3	0	125	10	0.1	2	34.36	34.09	0.8
4	200	100	13.5	1	1	83.2	83.0	0.2
5	105	100	50	1	6	1220.0	1191.8	2.4
6	0	100	50	1	4	577.6	561.3	2.7
7	0	900	9	2	1	129.8	125.7	3.3
8	0	1000	12.8	0.24	1	61.99	59.52	3.99
9	34	1600	42	0.65	3	1029.1	985.1	4.5

5.6 Models for Seasonal Demand

5.6.1 Review of Existing Models

The first model appeared in literature to deal with seasonal demand inventory problems is that of Hertz and Schaffir [147] which is very similar to the "news boy" problem. The basic criterion for deciding on the amount of inventory may be expressed in terms of the expected profit derived from each incremented unit of inventory. This expected profit is the difference between the estimated unit profit from sale during the regular end of the season and the estimated unit loss as a result of being left over at the end of the season, each weighted by its respective probability,

$$e_{ikw} = p_{ikw}c_i + (1 - p_{ikw})c_i'. \tag{5-90}$$

If the *i*th item is sold during the regular selling season, then the unit profit c_i is the difference between the regular selling price p_i and the variable cost r_i, that is

$$c_i = p_i - r_i \,, \tag{5-91}$$

and the expected unit lossc_i' based upon the reduced selling price p_i' will be

$$c_i' = p_i' - r_i . \tag{5-92}$$

p_{ikw} is the probability that the *k*th unit of item *i* will be sold by the end of the regular selling season. This probability is re-evaluated at every week during the season. Therefore, the optimal decision rule would be to order *k* units of item *i* for which $e_{ikw} \geq 0$. Obviously, it is too difficult to re-evaluate the probabilities p_{ikw} for every seasonal item at each week during the season. Moreover the personal experience and judgement about the practical situation is very important because the inventory decisions entirely depend on the estimated probabilities.

Another model for seasonal demand is presented by Levary [148] to minimise the total inventory costs during the seasonal cycle. It is formulated as:

$$Min\ \mathrm{Z(i)} = \sum_{t=i+1}^{M}\left[\left[\sum_{n}\mathrm{C}_{\mathrm{n}}(t)X_n(t)\right] + h(t)I(t)\right], \quad i=1,2,\ldots, M\text{-}1 \tag{5-93}$$

where

Cn(t) = unit variable cost of ordering from source n, during time period *t,*

h(t) = cost to store a unit from period *t* to period *t*+ *1,*

I(t) = inventory level at the end of period *t,*

M = numbers of periods during the time horizon,

Xn(t) = quantity of the substance, ordered from source *n*, during time period *t,*

Z(i) = total cost from period *i* + *1* to the end of considered horizon.

The uncertainty of seasonal movement is considered by defining some states of nature. Each state of nature has an estimated conditional probability of occurrence during each period within the seasonal cycle. The demand for an item can be estimated for each period with a known state of nature. Thus the expected demand during a period *t* is

$$D(i,t) = \sum_k p(k|k(1), k(2)\ldots., k(i)) D(k,i,t), \qquad (5\text{-}94)$$

where $D(k,i,t)$ is the demand for the item during the time period *t*, which is estimated at the end of period *i*, assuming a state of nature of *k* will occur.

This model has some serious drawbacks: (1) the order cost is not included in equation (5-93) which aims at minimising the total inventory costs; (2) the procedure to calculate the future expected demand is disconnected and diverted from the initial target to choose $X_n(t)$ to minimise $Z_n(i)$ because the expected demand in a period is decided by the state of nature and estimated demand (see equation 5-94), and this process can not guarantee minimum inventory costs as it is hoped; (3) the model is not practical because the user have to know the conditional probability of the states of nature and the corespondent demand at every period for obtaining the expected demand, in fact, they are very difficult to be estimated.

5.6.2 Analysis and Selection of Models

In general, the seasonal demand consists of a basic demand and a seasonal movement. The basic demand may be constant or has a trend, with a super imposed seasonal movement. Thus the seasonal demand can be expressed by:

$$D(t) = a + s_t + \varepsilon_t \qquad (5\text{-}95)$$

for the constant seasonality, and

$$D(t) = a + bt + s_t + \varepsilon_t \qquad (5\text{-}96)$$

for the trended seasonality respectively.

The fundamental work is to separate the seasonal movement, and then identify the pattern of the basic demand. The adopted inventory policies should be based on both the basic demand and the seasonal movement. In the following discussion, the lead time is assumed to be constant.

5.6.2.1 Separating the Seasonal Movement from the Seasonal Demand Pattern

The approach to separating seasonal movement from the seasonal demand pattern is a standard procedure used in forecasting [121]. If we have n observations from L seasons in m full years ($n = mL$) and n_i observations in the ith season, the seasonal movements with a constant basic demand can be obtained from:

$$s_i = \bar{y}_i - \bar{y} \qquad i=1,\ 2,\ \ldots,\ L \tag{5-97}$$

where

$$\bar{y} = \frac{\bar{y}_1 + \bar{y}_2 + \cdots + \bar{y}_L}{L} \quad and \quad \bar{y}_i = \frac{\sum_{i\,th\ \text{season}} y_t}{\mathrm{n_i}} \tag{5-98}$$

Similarly, the seasonal movement with basic demand having trend can be estimated from:

$$s_i = \bar{y}_{i(adj)} - \bar{y}_{(adj)} \tag{5-99}$$

where

$$\bar{y}_{i(adj)} = \bar{y}_i - \frac{(i-1)b}{L}, \quad \bar{y}_i = \frac{\sum_{i\,th\ \text{season}} y_t}{\mathrm{n_i}} \quad and \quad \bar{y}_{(adj)} = \frac{\bar{y}_{1(adj)} + \bar{y}_{2(adj)} + \cdots + \bar{y}_{L(adj)}}{L},$$

and b is the rate of change (slope) of yearly average demand.

5.6.2.2 Seasonal Variation Based on Constant Basic Demand

In this case, the seasonal demand can be described as a time independent constant rate plus a time dependent seasonal movement, i. e.

$$D(t) = a + x_1 s_1 + x_2 s_2 + \cdots + x_L s_L + \varepsilon_t$$
$$x_i = \begin{cases} 1 & in\ the\ ith\ season. \\ 0 & otherwise. \end{cases} \quad i = 1, 2, \ldots L. \tag{5-100}$$

If a year is divided into equal intervals according to the length of season, then the demand over each interval equals the sum of the constant demand and seasonal movement over the interval. This converts the problem into deterministic time varying demand inventory problems, the solution of which is reviewed in the following subsection.

5.6.2.3 Comparison of Different Models for Time Varying Demand

One of the first solutions to the above inventory problem is given by Wagner & Whitin [149] for the special case of pre-specified planning horizon (ending point). The mathematical model may be viewed as a "one-way temporal feasibility" problem (a forward path dynamic programming formulation), in that it is possible to order inventory in period *t* for demand in period *t+k*.

The functional equation representing the minimal cost policy for periods *t* through *N* is

$$f_t = \min_{\substack{Q_t \geq 0 \\ I+Q_t \geq D(t)}} [c_h I + \delta(Q_t)c_p + f_{t-1}(I + Q_t - D(t))] \qquad (5\text{-}101)$$

where

$$I = I_0 + \sum_{i=i}^{t-1} Q_i - \sum_{i=1}^{t-1} D_i(t) \geq 0,\ I_0 = \text{initial inventory, and } \delta(\mathrm{Q_t}) = \begin{cases} 0 \text{ if } \mathrm{Q_t} = 0 \\ 1 \text{ if } \mathrm{Q_t} > 0 \end{cases}$$

In period *N*

$$f_N(I) = \min_{\substack{Q_N > 0 \\ I+Q_N = D(N)}} [c_h I + \delta(Q_N)c_p]. \qquad (5\text{-}102)$$

Consequently, f_t is computed at *t* = *N* as a function of *I*, ultimately f_1 and an optimal solution are obtained as *I* for period *1* is specified.

The algorithm at period t^* (t^* = *1,2, ..., N*) can be generally stated as:

1. Consider the possible policies of ordering at period t^{**} (t^{**} = *1,2, ...,*t^*) and fulfilling the demand *D(t)*, $t = t^{**}, t^{**}+1, \ldots, t^*$, by this order;

2. Determine the total cost of these different policies by adding the ordering and holding costs associated with placing an order at period t^{**}, and the costs of acting optimally for periods 1 through t^{**}-1 considered by themselves. The latter cost has been determined previously in the computations for periods *t* = 1,2, ..., t^*-1;

3. Select the minimal cost policy for periods *1* through t^* considered independently;
4. Proceed to period t^* + *1* or stop if $t^* = N$.

This algorithm guarantees to provide a set of optimal reorder quantities that minimise the total inventory costs within a pre-specified horizon, but the relatively complex nature of the algorithm makes it difficult for the practitioner to understand. Although it sets up the foundation of the study of dynamic time varying demand inventory problem, it has received extremely limited acceptance in practice [36] because the optimal solution is based on the pre-specified ending point which is very difficult to predicate in practice.

Kaimann [150-152] compare Wagner-Whitin algorithm [149] with the use of EOQ and show that the Wagner-Whitin Algorithm provides substantial cost savings. But Tuite and Anderson [153] pointed out that an inappropriate cost expression had been used in Kaimann's analysis. In addition Gleason [154] and Gorrenstein [155] in their articles also point out that EOQ is not appropriate for this case because the conditions do not fit its assumptions and the computation reduction may not be worth the sacrifice of the optimal policy

Kicks and Donaldson [156] approximate the cumulative demand over *T* periods by $\frac{T}{2}\sum_{i=1}^{T} D_i(t)$. They have shown that the cost of ordering for *T* periods will be less than the cost of ordering for *T+1* periods if and only if

$$c_p / T + \frac{c_h}{2}\sum_{i=1}^{T} D_i(t) < c_p / (T+1) + \frac{c_h}{2}\sum_{i=1}^{T+1} D_i(t). \tag{5-103}$$

The reorder quantity $Q = \sum_{i=1}^{T} D_i(t)$ is determined by the smallest *T* for which the inequality $T(T+1)D_{T+1}(t) > 2c_p / c_h$ is true. The following reorder quantities are obtained by repeating the process starting at *i* = *T+1* and renumbering the periods *T* =

1,2, ... from that point. The principle merit of the above model is its simplicity, its shortcoming is that it does not guarantee an optimal solution because it involves two sources of error (1) it does not consider ordering for large number of periods which may have lower cost and (2) it uses an approximation for the cumulative demand over *T* periods.

Another important model is the Silver and Meal heuristic [157] that can be simply expressed as follows:

```
Step 1:
 Set $T = 1$;
 Go to step 2.
Step 2:
 If $T^2 D(T+1) > c_1 / pc_2 + \sum_{j=1}^{T} (j-1) D(T)$
    Go to step 3;
 else
    Go to Step 4.
Step 3:
 Set $T = T+1$;
 Go to step 2.
Step 4:
 $Q = \sum_{j=1}^{T} D(j)$.
```

The algorithm implies that the planning horizon is divided into equal intervals such as a week, a month or a quarter. The weakness of this heuristic is that it does not indicate how to decide on the basic intervals on which the optimal policy is based. Moreover, the model obtains the optimal replenishment quantity with the first increase in costs per unit time. In fact, it is possible that larger values of *T* would have lower costs per unit time when demand drops substantially in period *T+2*.

Despite the complex calculations, Wagner-Whitin [149] algorithm demands the restricted condition that the demand of every interval over the planning horizon must be known exactly and unchanged which seems to be impossible in practice. Instead, the Silver and Meal [157] heuristic of selecting replenishment quantities for the case of time varying demand is very simple and easy to implement. It can also cope with the future fluctuation of demand in each period within the planning horizon (in fact, Silver and Meal heuristic does need an ending point). In addition, the use of Silver and Meal heuristic can only result in an average cost penalty of 0.4% with the optimal algorithm [153], thus the Silver and Meal heuristic is selected for the proposed system.

5.6.2.4 Seasonal Variation Based on Basic Demand Which Has a Linear Trend

In this case, the demand consists of a basic linear trend and seasonal movement. Thus the replenishment decision is made following two stages. First, to decide the replenishment interval T of the basic component as the basis of reorder interval; then add the seasonal movement for this interval. Thus the problem can be solved by deciding the replenishment quantity for each demand component, i.e. the basic component with linear trend and the seasonal movement component during the interval. The first component can be determined by Mitra's et al [145] model (see Section 5.4.2). The seasonal movement (Q_s) within replenishment quantity can be decided in terms of the length of replenishment interval T and the length of season l. If T/l is an integer, the seasonal movement is

$$Q_s = \sum_{i=1}^{\frac{T}{l}} s_i \, . \tag{5-104}$$

If T/l is not an integer, it is separated into an integer n and a fraction α ,

$$T/l = n + \alpha \ . \tag{5-105}$$

and the seasonal movement is decided by

$$Q_s = \sum_{i=1}^{n} s_i + \alpha \cdot s_{i+1} \, . \tag{5-105}$$

The seasonal movements are obtained from the demand data of the past few years based on the premise that the seasonal fluctuation will repeat in the future. In practice the present circumstance could be different from the past so that the seasonal movement calculated using the history data is only indicative. Thus the user will have to amend it by taking the present condition into account.

5.7 Summary

In this chapter the models included in the knowledge base of the proposed system are discussed. These models cover the range of inventory problems of constant demand, probabilistic demand, low demand items, demand with linear trend, and seasonal demand under the conditions of single item and a single location.

The models which deal with the each inventory problem are compared and the most suitable models have been selected for making reordering decisions and calculating the order quantity. Suitability of models is based on its applicability, easiness of understanding, and being adaptable to a computer solution.

Gaps have been found in literature where models for some situations have not yet been published. This prompted the need to develop these models which mainly concern with situations of constant demand with discount, shortage, and restricted supply rate. We also modified Mitra's et al [145] model for demand with linear trend to cope with the situation where the number of reorders $n = D / D^{*}T^{*}$ is not an integer and proposed a method to deal with seasonal demand with basic demand element that is either constant or has a trend.

Chapter 6

THE DEVELOPMENT OF THE INTELLIGENT INVENTORY SYSTEM

This chapter describes the development of the proposed system following the overall structure outlined in Chapter 3 and incorporating the models discussed in Chapter 5.

6.1 The Environment of Developing the System

Having chosen the structure of the intelligent inventory system, consideration must now be given as to how to develop this application. The requirements for developing any computer applications can be divided into two broad areas: hardware and software. This section defines these terms, and aims to give an insight into some of the different available options.

6.1.1 Software

There are basically two types of expert systems development tools. These tools are the knowledge engineering languages/systems and the high-level programming languages.

Knowledge engineering languages/systems (also known as shells) are sophisticated expert system building tools which are comprised of an expert system building language and an extensive support environment.

The advantage of the knowledge engineering languages/systems is that they are easier for the inexperienced expert system builder to apply. The shell essentially dictates how knowledge is to be represented and contains a pre-constructed inference engine for accessing the knowledge base. Although the names of knowledge engineering shells are not as well known as those of standard high-level programming languages, shells have become increasingly popular in recent years.

Despite their popularity, shells have some disadvantages. As the inference engine of a shell is already structured, the range of problems that can be solved in a given domain is limited by the constraints enforced by the shell. The limited number of users also stagnates the development of KBS shells and results in higher prices of these shells. In addition as more and more shells are continually being developed, software houses that develop these tools are often in a big hurry to get the tools to the market place and so testing of their reliability is sometimes underestimated.

On the other hand, high-level programming languages that are used in developing expert system applications include symbolic-manipulation languages such as LISP and PROLOG and procedure oriented languages such as Pascal and FORTRAN. Most of these high-level languages are mature, and for some of them there are a number of support facilities available such as debuggers, graphics interfaces, and full-screen editors. These facilities are helpful to the programmer. There are advantages and disadvantages to the use of these high-level programming languages.

The main advantage of the high-level languages is that they allow a great deal of flexibility for the expert system builder. Because of this flexibility, an expert system built with these tools can usually solve a wider variety of problems within the same domain. Their disadvantage is that these languages are more difficult to apply because they give little or no guidance on how knowledge should be represented or how mechanisms for accessing the knowledge base should be designed

The proposed inventory IDSS system has been developed using Microsoft Visual Basic for Windows. Visual Basic was selected as the development tool because of its significant advantages. Apart from being faster, more compact and as portable as other traditional languages are, Visual Basic provides many tools and includes an easy-to-use language that is compatible with QBasic and other Microsoft Basic products. Moreover, it can also communicate with other Windows applications and call executive files developed by any traditional language. This will make it easy to interface the system with most of the existing commercial systems as they are usually written by traditional languages.

The latest development of the Visual Basic is the package M.4 VB [158] which is a windows based expert systems development tool. It provides Visual Basic users with the ability to add an experts' knowledge to their applications. This new subset of the complete M.4 VB package is specifically targeted towards Visual Basic application developer, and provides a custom control with complete sources for all types of interface. Both procedural control and object oriented programming are integrated with a complete knowledge base development environment and language. Dozens of inputs can be entered on a form and then intelligently processed by M.4 VB all at once. This new version of Visual Basic combines the advantages of both the shells and the high-level languages.

The choice of the right tool is not always an easy task [71, 75]. The initial choice of a particular development tool may need to be changed because it is simply not doing what the knowledge engineer anticipated. This potential problem is likely to occur to inexperienced knowledge engineers. With the experienced knowledge engineers, it is often the case that the engineer has already developed a preference for one particular tool or set of tools which may not be a good match to the application to be developed.

The first consideration when choosing the development tool is the form of input and output. The Visual Basic have an adequate set of support facilities such as forms, controls, icons, and menu design which are major assets to the development process. By combining the Visual Basic forms, controls, and programming language one can build a graphical user interface quickly and easily. As we have already pointed out, the user interface can determine whether the application is accepted and utilised.

The second consideration when choosing the development tool is the requirement of the application. The approach of the implementation of the proposed system adopted in this thesis is the integration of the pattern identifier with the knowledge base. The pattern identifier (see Section 3.3.3) is used to trace history data and predict necessary information such as a demand pattern which is essential to the rule base to make inventory management decisions. This pattern identifier carries out complicated calculations which can be handled more efficiently by the high-level languages than by shells.

The finial consideration of choosing the development tool is how to integrate the proposed system with other applications in order to utilise existing data/information. The Visual Basic provides the facility to access any windows applications. The business world over the past decades has stored tremendous amounts of information in computer-based database and spreadsheet files which may be on mainframes or microcomputers. It is not surprising , therefore, that one of the most current trends in KBS design is the integration of IDSS with a spreadsheet format.

6.1.2 Use of Visual Basic in System Development

The Visual Basic is a complete event-driven programming language that supports the structured programming constructs found in most other modern programming languages. It is substantially different from traditional BASIC, though there are similarities. The Visual Basic programming system allows users to create attractive and useful applications that fully exploit the graphical user interface (GUI) and the key features of Microsoft Windows, including multiple-document interface (MDI), object linking and embedding (OLE), dynamic data exchange (DDE), graphics, and etc. The Visual Basic can also be extended by adding custom controls and by calling procedures in dynamic-link libraries (DLLs). A Visual Basic application usually consists of the following essential components (Fig. 6.1):

1. Project. A project is the collection of forms, modules, and controls used to build an application.

2. Form. A form is a window which is a container for controls.

3. Control. A control is an object that has a set of properties and one or more events it recognises.

4. Procedure. A procedure is a collection of codes.

5. Event. An event is an action recognised by a form or control. Event-driven applications execute code in response to an event.

Each form and control in Visual Basic has a pre-defined set of events. If one of these events occurs, Visual Basic invokes the code in the associated event procedure. This is the essence of graphical user interfaces and event-driven programming. When all the files for a project have been created, it can been converted into a executable file with extension *.exe*. The forms and modules developed in the system described in this thesis are listed in Table 6.3 and Table 6.4 respectively.

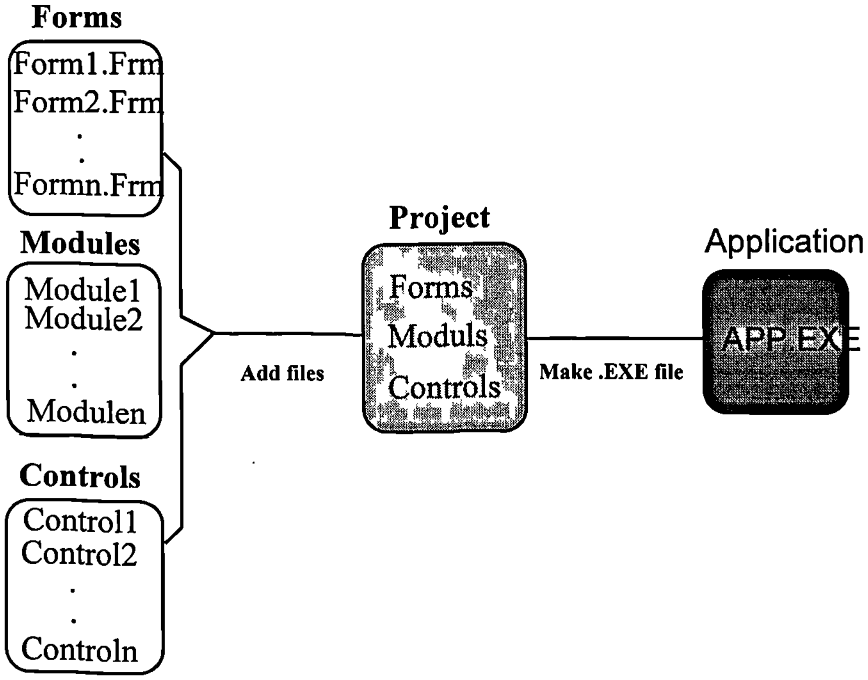


Figure 6.1 The Structure of a Visual Basic Application

Using Visual Basic 3.0 to develop an application, the following limitations should be taken into account [160]:

1. There is a limit of approximately 230 forms in a project and an application can have up to 80 of these forms loaded at one time.

2. The sum of all procedures, modules and forms, and DLL declarations in the application must be less than 5,200.

3. The maximum number of controls allowed on a single form is 470 and the maximum number of controls in all open forms is about 600.

4. Under Windows 3.1, the total number of windows that can be in existence at one time is about 600 windows; under Windows 3.0, this limit is about 475 windows.

6.1.3 Hardware

Hardware is the computer equipment and its peripherals, which make up the computer system. Peripherals include keyboards, visual display units, printers etc.

Computers can be divided into three categories: mainframe computer, mini-computer and desktop computer. The mainframe is a large multi-purpose computer usually serving the needs of a large community. The desktop computer or Personal Computer (PC) is the smallest of computers and it is a single user machine. There is a wide variety of software available to provide sophisticated facilities for the PCs. The network (or distributed system) consists of two or more computers connected to exchange data or share resources such as hard disc space, program and data, printers, and other devices.

The hardware supporting the development of the proposed system is a PC with 486 processor. One reason to choose PC to support the development of the system is due to the fact that the Visual Basic can run on the Microsoft Windows 3.1 that is available in the Novell Network installed at the University of Salford. Another consideration is that the PCs are widely used in business.

6.2 The Development of the User Interface

6.2.1 Overview of the User Interface

The user interface is the lifeline of an expert system. It is the total communications channel between the user and the expert system [71, 72]. If an expert system is perceived as a circle, then the surroundings of the circle is the user interface as shown in Fig. 6.2.

If the user interface is appropriate for the intended user, then the users will be able to communicate effectively with the expert system. This communication is considered to be not only the physical method of sending and receiving information from the expert system, but also the perceptual and psychological media. The user's perceptual image of the system is critical to the overall success of the system. If the tasks, functionality, and benefits are not readily understood by the user, then the system will not be effective. Moreover, from a psychological point of view, if the user perceives the system as confusing because of lack of proper instructions, then the user interface is inadequate.

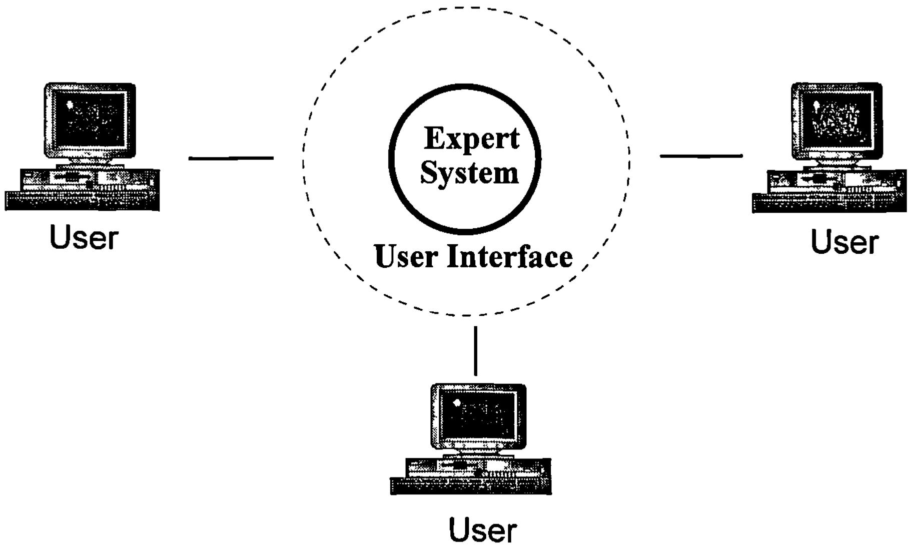


Figure 6.2 Illustration of the User Interface

Given the importance of the interface to acceptance of the expert system, it is not surprising that the burden on designing the interface has greatly increased. The interface design is more complicated than just putting up the windows on the screen. The user interface should include several important capabilities such as the data captured from the user, data presentation through a range of output devices, help facilities, windowing, graphical presentation and providing data exchange channels with other applications [159].

6.2.2 Considerations when Designing an Expert System's Interface

6.2.2.1 The Demand on the User Interface

Compared with the behaviour of a traditional software system, an expert system is not just a tool that implements a process, but rather it is a representation of that process. The user interface must often present not only conclusions, but an explanation of the processes by which those conclusions are reached. In this system, different explaining facilities such as reporting windows, message boxes, and displaying windows have been developed to explain system behaviour and display the data that will convince the user of the effectiveness of the system.

6.2.2.2 Compatibility with Other Interfaces

Another important factor to be considered is the users' environment. When asked what kind of interface the users prefer, many users prefer the interface design of the first computer application they learned because very few users in the business environment are exposed to all kinds of interfaces in their daily routine [71]. If the users are unfamiliar with the interface, they may be resistant to the amount of learning required to become proficient at handling the new interface. Therefore, factors such as the users' access to the system, their time to use it, and the compatibility with other interfaces that the users have been exposed to must be taken into account.

The user interface of this system is highly graphical and similar to other windows applications because Windows environment has established itself as the standard screen format for business software interfaces.

6.2.2.3 Fitting into the Data Environment

This refers to the integration of an expert system with the users' existing data environment. If the users have spent much time entering data into an existing database system, it would not be wise to require them to re-enter much of this data to the new system. Therefore, it is crucial that such information can be captured by the new system. Also the results generated by the new system have to be transportable to other applications for further processing.

The user interface of the proposed system can exchange data with Excel which is a widely used application and access any system developed by using traditional languages.

6.2.3 Screen Design

The user's perceptual image of the user interface is the screen layouts [71]. There are many different types or styles of interface designs in modern expert systems. The following common screen layouts and types of interactions, ported from the trends in microcomputer software applications development, have been adopted in the interface design.

6.2.3.1 Question and Answer (Q/A) Box

It is the simplest and earliest format of screen lay out. The Q/A box (Fig. 6.3) displays a question on the screen and leaves an empty text box for the user to type in a reply with the keyboard or any other device. Then the expert system takes the reply and incorporates the information within the knowledge base. This method is quite intuitive. The disadvantage is that it allows the user to retype or re-enter incorrect or misspelled answers.

6.2.3.2 Menus

A menu-driven interface displays a list of choices to the user as shown in Fig. 6.4. The menu bar is arranged horizontally (hierarchical format) for the user to move across the menu bar to highlight a menu. The choices (or commands) of the menus are arranged in a vertical format (pull-down or pop-up). The user's selection could lead to another submenu. A triangle (▸) next to a command indicates a cascading menu listing the additional commands if this command is selected. An ellipsis (...) following a command means a dialogue box will appear when the user choose the command. Both of these menu formats, hierarchical and pull-down, have the advantage of requiring less area on the screen and they are also familiar to most office workers and professionals who are already using them on a daily basis. However, both require considerably more screen programming to be developed.

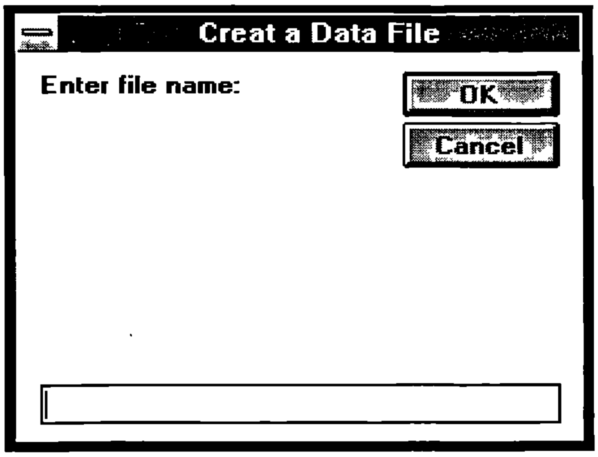


Figure 6.3 Question and Answer Type Screen Layout

6.2.3.3 Windows

The latest generic screen design is called the windowing environment [160]. Windows are rectangles or frames of information that appear on the screen which consist of Control Menu Box, Tittle Bar, and Resize Buttons to allow the user to view, edit, rearrange, enlarge, and perform similar manipulations on windows (Fig. 6.5). Windows are the main screen configuration of the proposed system's user interface.

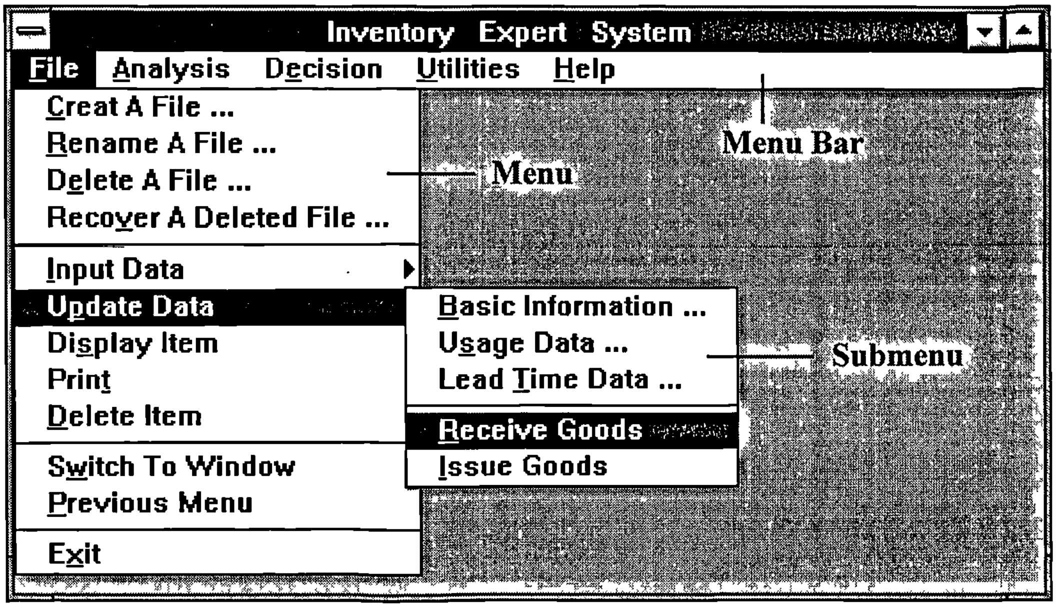
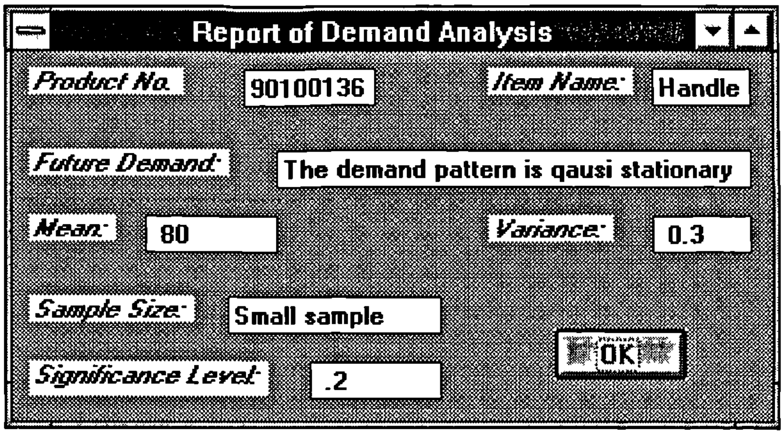


Figure 6.4 Menu Type Screen Layout



Product No. 90100136

Item Name: Handle

Future Demand: The demand pattern is qausi stationary

Mean: 80

Variance: 0.3

Sample Size: Small sample

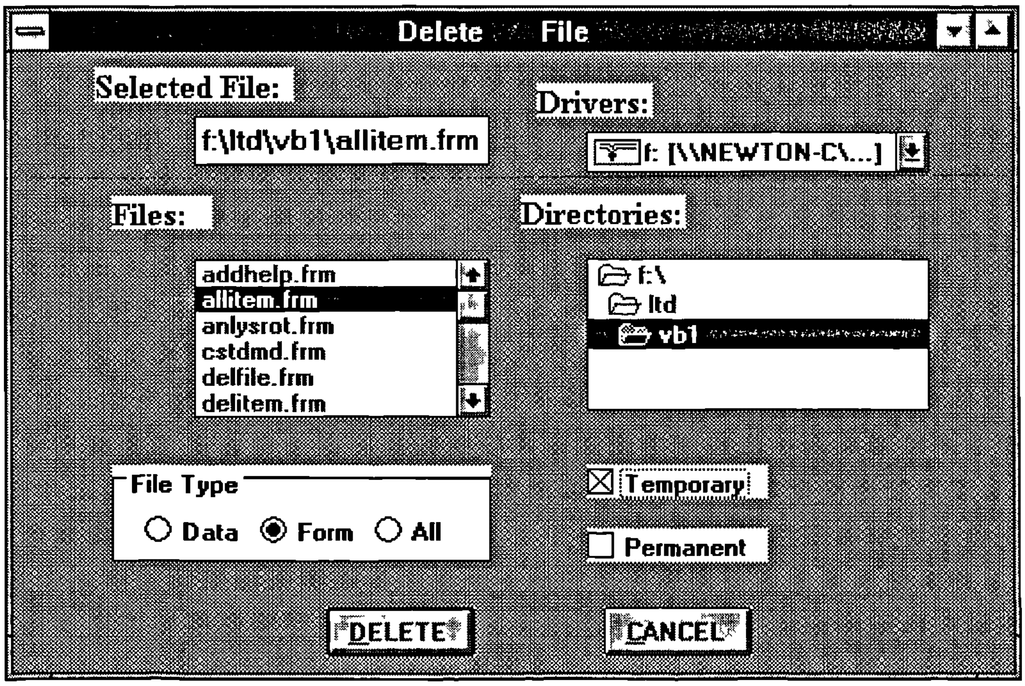
Significance Level: .2

Figure 6.5 An Example of Windows

One kind of special windows called Dialogue Box [160] (Fig. 6.6), is a window where users actually enter information about a task to be performed. A Dialogue Box often contains the following objects:

- Text Box accepts user's replies to the system.
- List Box lists of available choice appears.

- Option Buttons present mutually exclusive options.
- Check Boxes offer multiple options for user to select.
- Command Buttons initiate an immediate action.



Selected File: f:\ltd\vb1\allitem.frm

Drivers: f: [\\NEWTON-C\...]

Files: addhelp.frm, allitem.frm, anlysrot.frm, cstdmd.frm, delfile.frm, delitem.frm

Directories: f:\, ltd, vb1

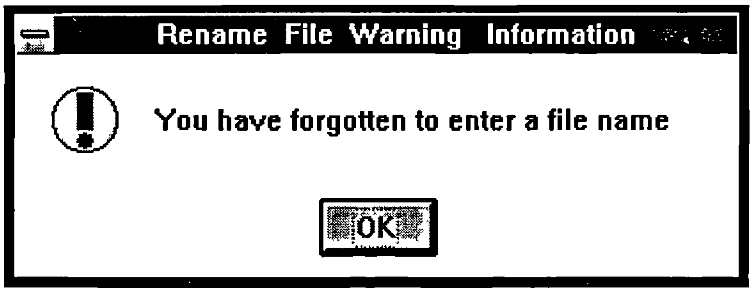
File Type: ○ Data ● Form ○ All

☒ Temporary

□ Permanent

Figure 6.6 An Example of Dialogue Box

Another kind of special window called Message Box [160] (Fig. 6.7), may appear when the system wants to alert the user to unusual situations or to ask the user to confirm unrecoverable actions. Table 6.1 shows the different types of information that could be displayed in a message box.



You have forgotten to enter a file name

Figure 6.7 An Example of Message Box

The advantage of dialogue box and message box is that the multiple-information boxes (windows) can appear simultaneously on the screen. The recently developed tools have overcome its disadvantage of requesting expertise to create the windows on the video system.

The user friendly interface of the system has been created using the above mentioned elements. All these screen configurations have been mixed together to develop the system's interface.

Table 6.1 The Types of Information Given by the System

Type	Purpose	Example
Prompting information	Tell the user what the system is doing	"Reading data ..."
Confirming information	Confirm an unrecoverable action	"Do you want to delete the file"
Warning information	Remind an inappropriate operation	"Forgotten enter a file name"
Error information	Prompt what kind of errors occurred	"Divided by zero"
Interim result	Display the interim result	"Trend demand hypotheses rejected"
Final result	Display the final result	"All the data have been transferred"

6.2.4 Screen Design Guidelines

During the development of the screen layout, the following directions were followed:

1. The objects were arranged on windows with appropriate spacing between them.

2. The wording and locations of prompts, labels and other identifiers were consistent throughout the interface.

3. Data entry field (text box) is highlighted using different colour to draw attention to a particular field when the user is filling a form-like screen.

4. The input data and text are left justified to facilitate the conversion between different data types.

6.3 Data Manager

This module is used to manipulate the data used to generate some parameters for model selection and calculate the optimal order quantity.

6.3.1 Data File Management

This sub-module, like the file menu of most of the business applications, has the capability to create, rename, delete, and recover a deleted data file. These commands are listed on the top section of the system's file menu. To reduce the user's typing errors, most of the available inputs options are listed on the dialogue box for the user to select. Fig. 6.6 shows the dialogue box of "delete file" command. The existing directories and files are displayed in the middle of the dialogue box. Therefore, the user can choose the file to be erased from the file list instead of typing in the file name. The processing details of this sub-module is illustrated in Fig. 6.8.

6.3.2 Data Record Management

The data record consists of different fields which represent an inventory item's properties such as holding cost, ordering cost, demand patterns, lead time distributions, etc. All of the items' records are stored in a data file with same data structure.

This sub-module includes the functions of capturing data from the user (or from other application), modifying data record of an inventory item, displaying and printing the data record (analysis results) stored in the data files, and deleting a data record from the data files. The flow chart of this sub-module is shown in Fig. 6.9.

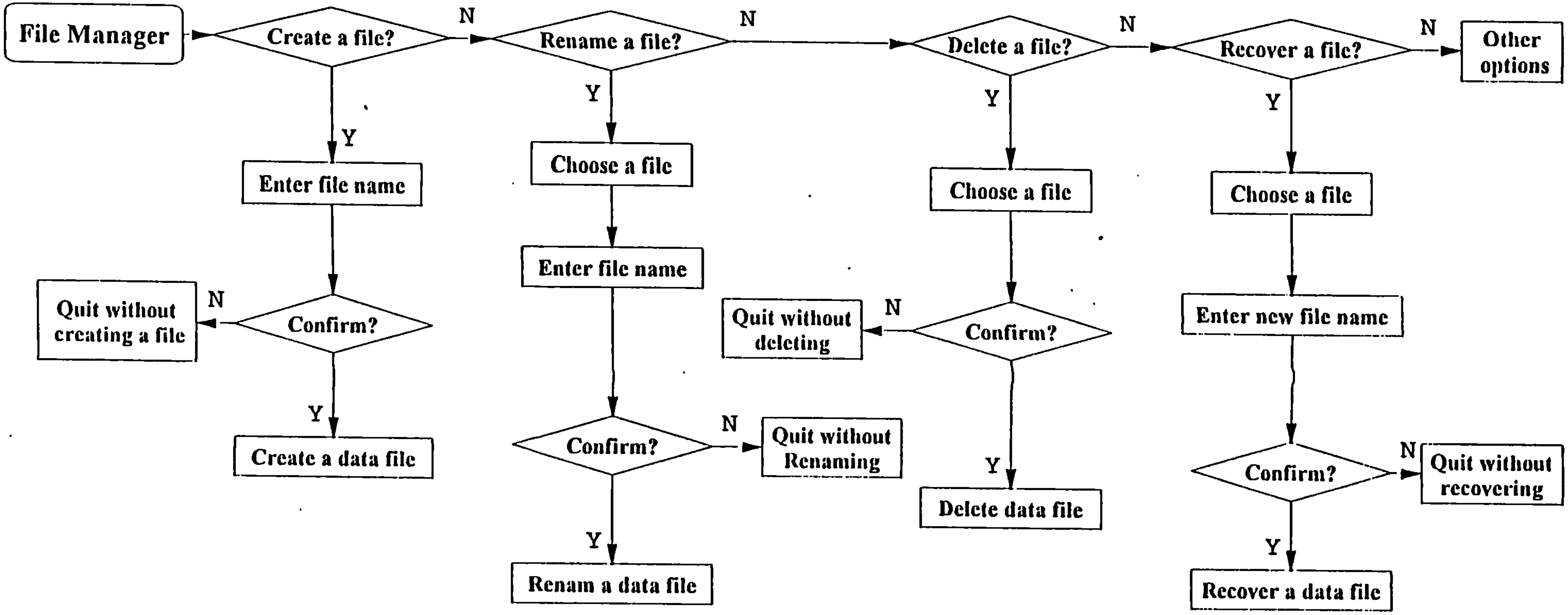


Figure 6.8 Flowchart of File Management

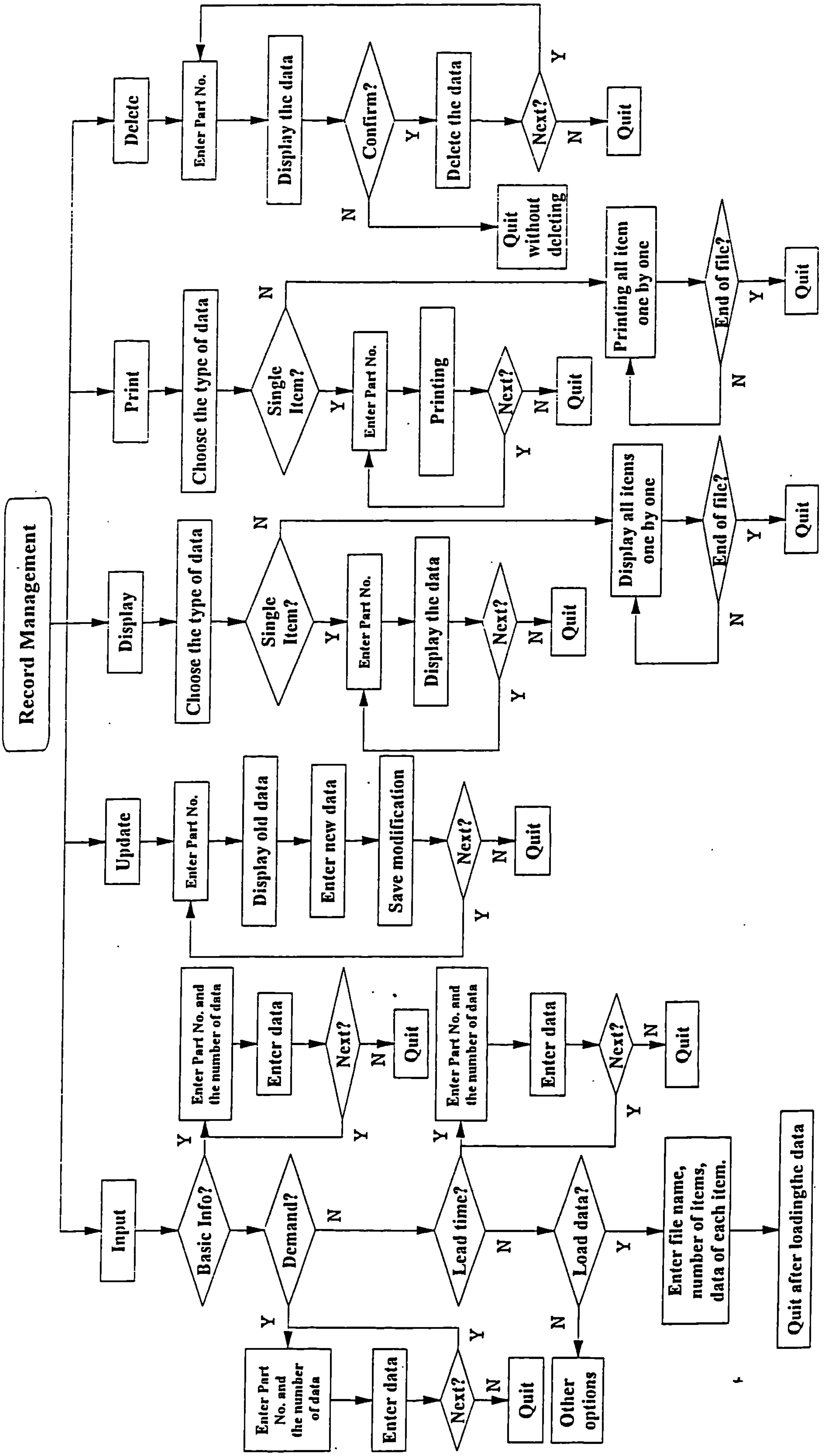


Figure 6.9 Flowchart of Record Management

6.3.3 Exchanging Data with Other Applications

Visual basic applications run in a multi-tasking operating environment along with other applications. Therefore, it is very easy to extract data from other applications using dynamic data exchange (DDE) to avoid re-entering the data which is stored in an existing system and thus reducing the typing errors. The DDE is a mechanism supported by the Microsoft Windows operating system that enable two applications to "talk" to each other by continuously and automatically exchanging data. The application which initiates the conversation is called destination application; the application responding to the destination is the source application. A DDE conversation is often called a link because the two applications in the conversation are linked by the data by which they are exchanged. There are three kinds of links distinguished by how the source updates the destination when data in the source changes:

1. Automatic link. The source supplies data to the destination every time the data defined by the linking item changes.

2. Manual link. The source supplies data only when the destination requests it.

3. Notify link. The source notifies the destination when data changes, but supplies data only when the destination request it.

To begin a DDE conversation, the destination must specify three things:

1. Application Name. Every application that can be a DDE source has a unique application name. Usually this is an executable file name without any extension (Table 6.2).

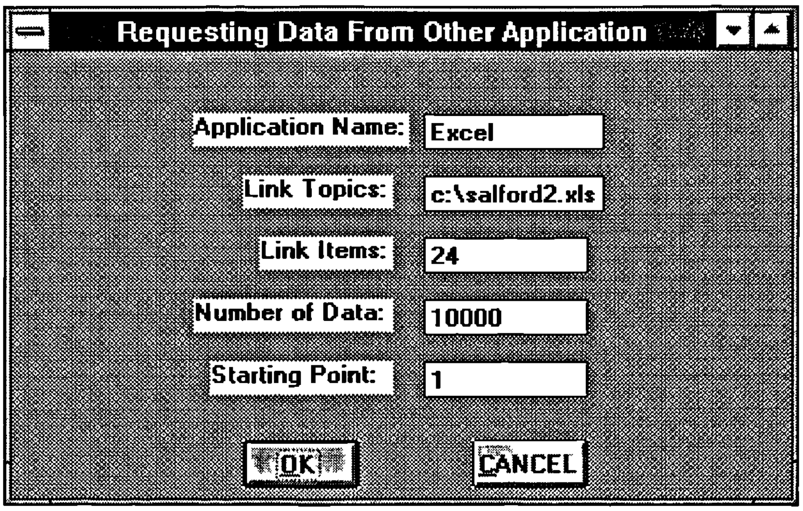
2. Link Topic. The topic defines the subject of a DDE conversation and is usually some unit of data that is meaningful to the source application. For example, Microsoft Excel recognises a file name ending in .XLS as a topic for DDE conversation.

3. Link Item. The item defines the piece of data actually being passed during a DDE conversation. For example, Excel recognises cell references (such as R1C1) as item in a conversation.

Table 6.2 The Application Names of Some Microsoft Applications

Application	DDE application name
Microsoft Excel	Excel
Microsoft Word for Windows	Winword
Microsoft Project	Project
Microsoft Access	MSAccess
Microsoft FoxPro for Windows	FoxPro

If the application name, link topic, and link item are specified in the programming code, it can only extract data from a particular application. In order to access other applications in which the data were stored, the code for requesting data has to be rewritten which is very difficult for the users. In this system, this procedure has been designed for fairly general purpose, i.e. instead of coding them into the procedure the application name, link topic, and link item are optionally entered by the user in according to where the data come from (Fig. 6.10). This means that the system can theoretically exchange data with any application which can serve as a conversation source.



Requesting Data From Other Application

Application Name: Excel

Link Topics: c:\salford2.xls

Link Items: 24

Number of Data: 10000

Starting Point: 1

Figure 6.10 The Data Exchange Window

6.4 Patterns Identifier

This module is one of the most important modules of the system. It is the implementation of the statistical methods described in Chapter 4. The module consists of two sub-modules which are used to identify the demand and lead time patterns.

6.4.1 Identifying Demand Patterns

This sub-module can identify the demand patterns of inventory items including seasonal movement, linear trend pattern, constant pattern and probabilistic pattern. The flowcharts of these procedures are show in Fig. 6.11, Fig. 6.12, Fig. 6.13, and Fig. 6.14 respectively. This sub-module is designed to identify the demand pattern of one item or identify the patterns of all of items in a file. In the one item case, some intermediate and final results are reported to the user. The intermediate results are not reported in the case of analysing a number of items to save computer running time. This element is the key feature of the system which is not attempted in all published inventory IDSSs.

6.4.2 Identifying Lead Time Patterns

There are two different methods that have been employed in this sub-module. If the lead time history data are available, the statistical methods (same as those used for identifying demand patterns) are used to identify the lead time pattern. Otherwise, the lead time pattern is obtained by expert subjective estimation. Fig. 6.15 shows the procedure of identifying the patterns of lead time.

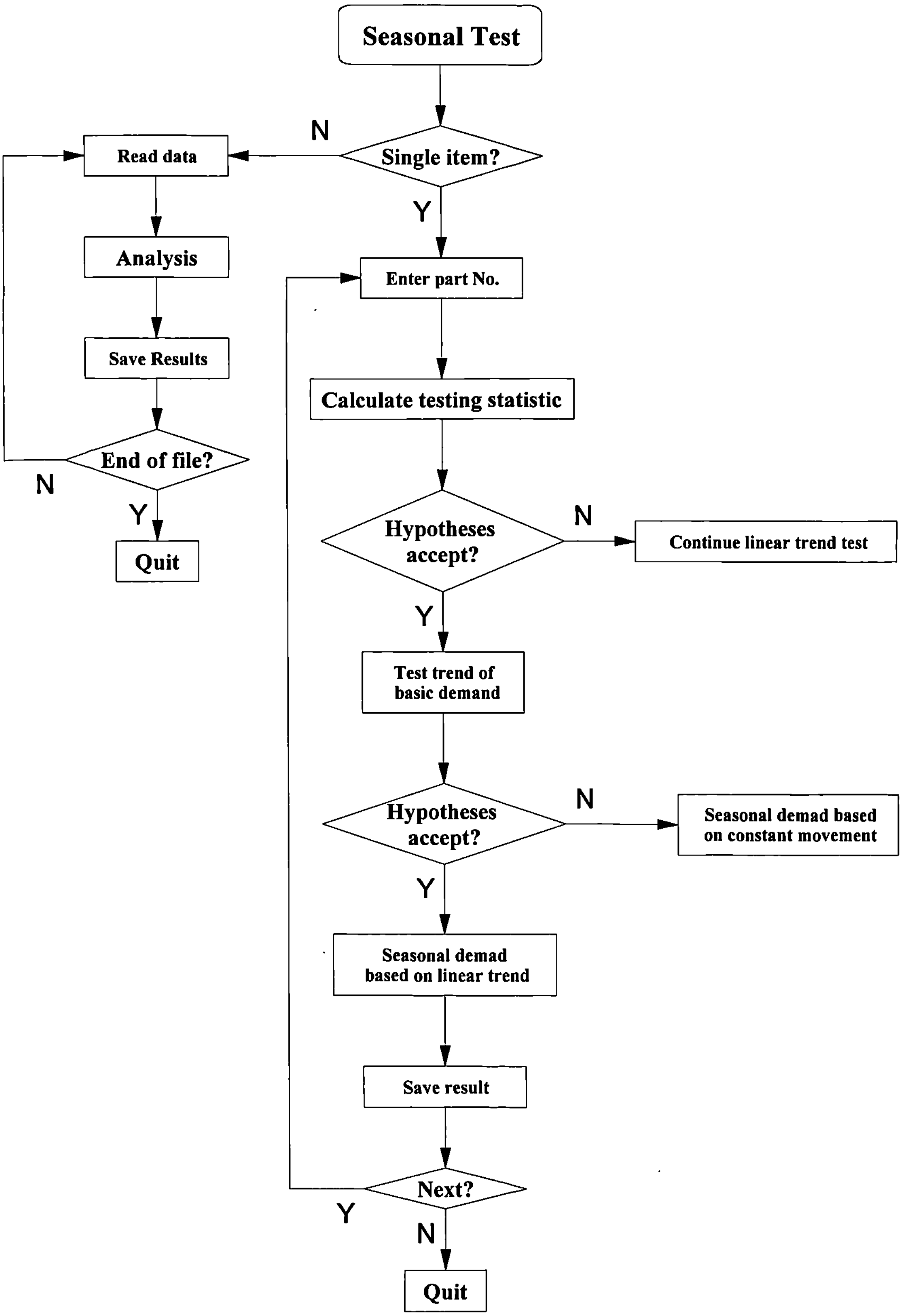

Seasonal Test
Single item?
N
Y
Read data
Analysis
Save Results
End of file?
N
Y
Quit
Enter part No.
Calculate testing statistic
Hypotheses accept?
N
Continue linear trend test
Y
Test trend of basic demand
Hypotheses accept?
N
Seasonal demad based on constant movement
Y
Seasonal demad based on linear trend
Save result
Next?
Y
N
Quit

Figure 6.11 Flowchart for Identification of Seasonal Demand Pattern

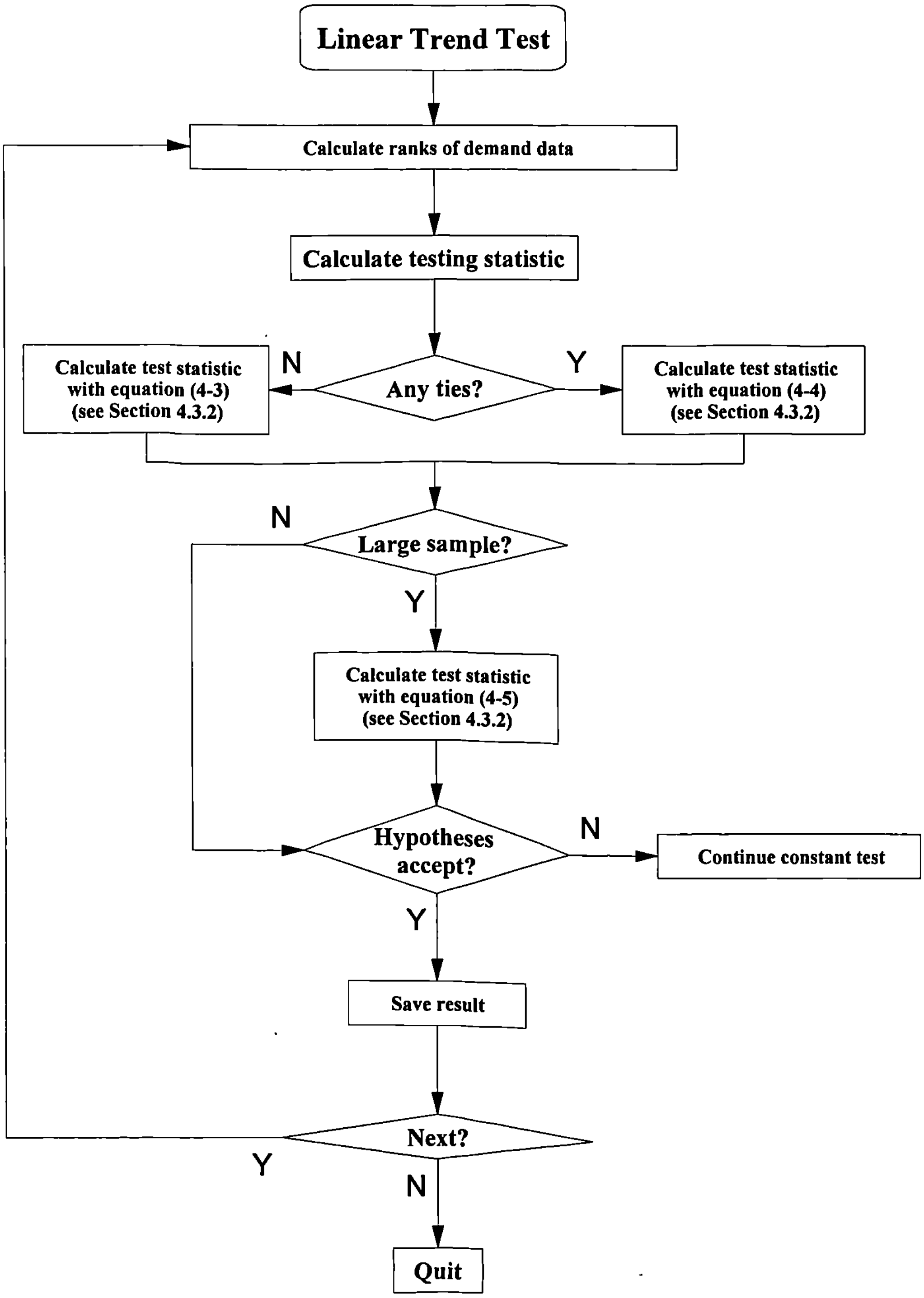

Linear Trend Test
Calculate ranks of demand data
Calculate testing statistic
Any ties?
N
Y
Calculate test statistic with equation (4-3) (see Section 4.3.2)
Calculate test statistic with equation (4-4) (see Section 4.3.2)
Large sample?
N
Y
Calculate test statistic with equation (4-5) (see Section 4.3.2)
Hypotheses accept?
N
Continue constant test
Y
Save result
Next?
Y
N
Quit

Figure 6.12 Flowchart for Identifying Demand Pattern with Linear Trend

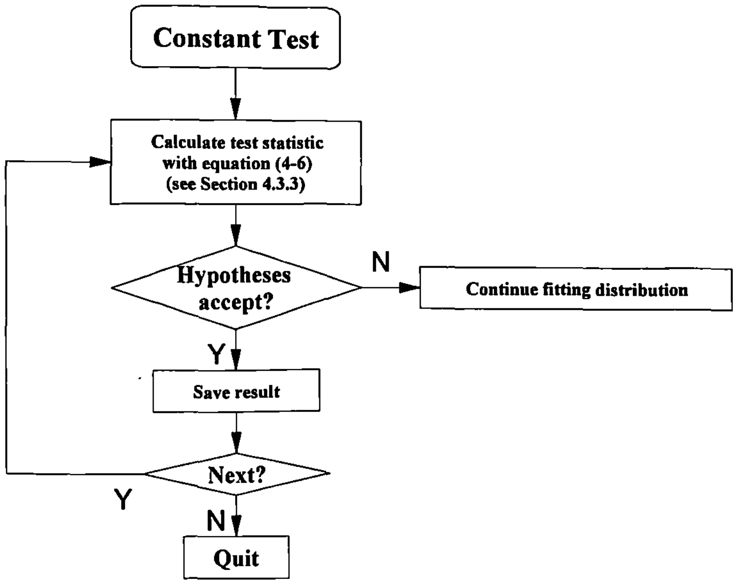
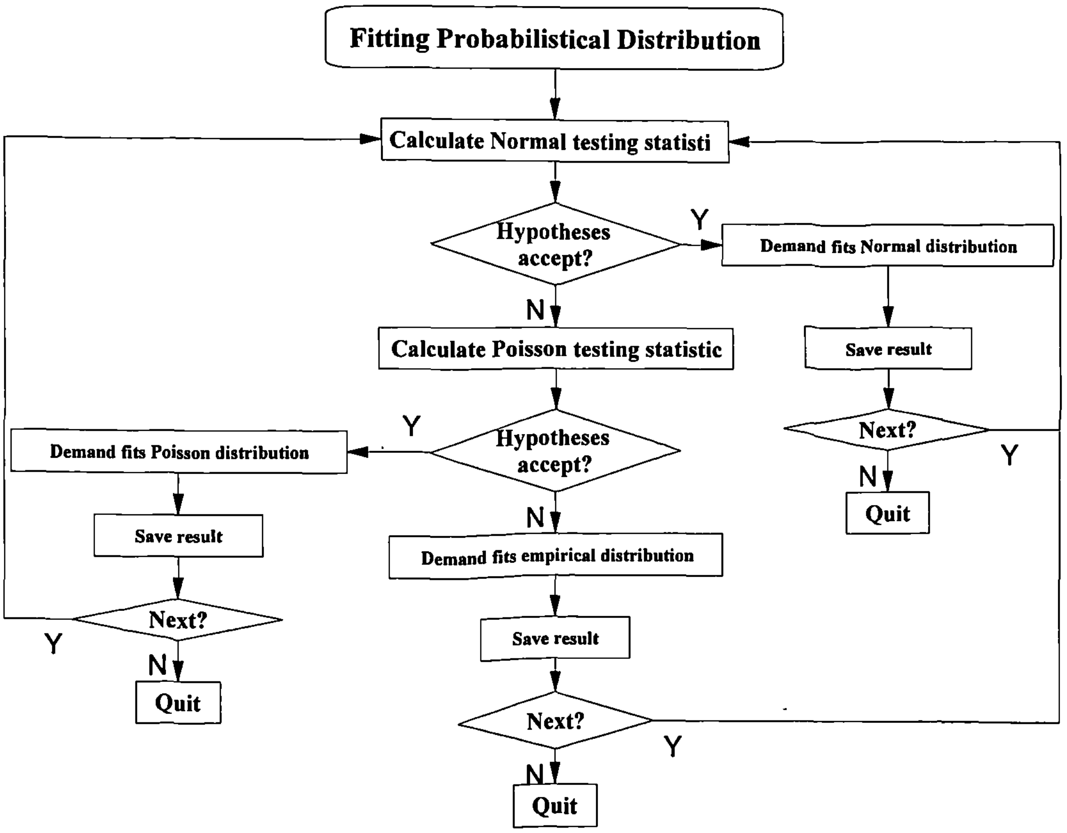

Constant Test
Calculate test statistic with equation (4-6) (see Section 4.3.3)
Hypotheses accept?
N
Continue fitting distribution
Y
Save result
Next?
Y
N
Quit

Figure 6.13 Flowchart for Identifying Constant Demand


Fitting Probabilistical Distribution
Calculate Normal testing statisti
Hypotheses accept?
Y
Demand fits Normal distribution
N
Calculate Poisson testing statistic
Save result
Y
Demand fits Poisson distribution
Hypotheses accept?
Next?
N
Y
Quit
Save result
N
Demand fits empirical distribution
Next?
Y
N
Quit
Save result
Next?
Y
N
Quit

igure 6.14 Flowchart for Identifying and Fitting Probability Distribution to Stochastic Demand Pattern

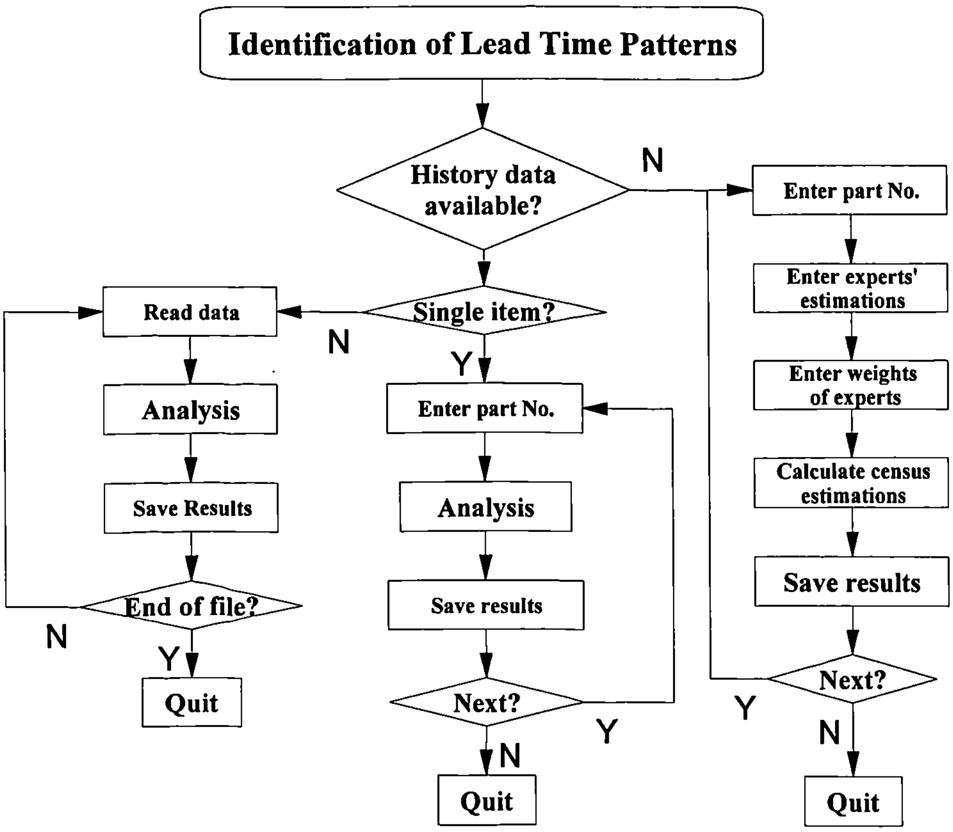


Figure 6.15 Flowchart for Identification of Lead Time Pattern

6.5 Model Base

The system described in this thesis incorporates a set of mathematical inventory models (discussed in Chapter 5) which enable the user to reach optimal ordering decisions. From the point of view of computer-based systems, the models can be viewed as subroutines, data or statements [161]. However, a more restricted viewpoint treats models as computerised procedures or executable routines [162]. A more conventional approach is adopted in developing this system by coding models as programme subroutines. They are invoked by the rules which select an appropriate model for the given conditions. The order quantity decisions are obtained by calling the appropriate model subroutine which has arguments representing the demand parameters.

This system was designed with modular structure. This modularity may be required to extend the model base and the rule base. Adding an additional model, simply requires the inclusion the new model's program code to the model base subroutines, and the addition of an appropriate selection rule to the model selection rule base.

6.6 The Model Selection Rule Base

The choice of the most appropriate model to apply to a situation represented by real-world data requires expertise. This is provided by the knowledge base or the rule base of the inventory IDSS.

Representing knowledge in the form of the procedural rules is by far the most commonly used knowledge representation technique in operational research applications [68]. The rapid development, efficiency of execution and ease of use and integration offered by this approach make it the preferred option for this system. A further key benefit is that rules are easy to communicate to domain experts.

The rule base of this system consists of a list of rules capturing some of the knowledge of experts in inventory modelling concerning mathematical modelling techniques and their applicability to various situations.

In this system the rule base was coded as procedural rules in one module to facilitate the revision and enhancement of the rule base by the system developers or users. The following set of rules has been implemented in this system. The system's reasoning outputs will not be affected by the order of how these rules be implemented because these rules are independent to each other. However, the reasoning strategy can be influenced by meta-rules which direct the search to specific portions of the KB [70].

6.6.1 Rule for Model Selection

Rule 1 If the lead time is constant
and the demand is constant
and shortage is not allowed
and supply rate is infinite
and no discount
Then apply the constant model Q1

Rule 2 If the lead time is constant
and the demand is constant
and shortage is not allowed
and supply rate is infinite
and all units discount
Then apply the constant model Q2

Rule 3 If the lead time is constant
and the demand is constant
and shortage is not allowed
and supply rate is infinite
and increment discount
Then apply the constant model Q3

Rule 4 If the lead time is constant
and the demand is constant
and shortage is not allowed
and supply rate is finite
and no discount
Then apply the constant model Q4

Rule 5 If the lead time is constant
and the demand is constant
and shortage is not allowed
and supply rate is finite
and all units discount
Then apply the constant model Q5

Rule 6 If the lead time is constant
and the demand is constant
and shortage is not allowed
and supply rate is finite
and increment discount
Then apply the constant model Q6

Rule 7 If the lead time is constant
and the demand is constant
and shortage is allowed
and supply rate is infinite
and no discount
Then apply the constant model Q7

Rule 8 If the lead time is constant
and the demand is constant
and shortage is allowed
and supply rate is infinite
and all units discount
Then apply the constant model Q8

Rule 9 If the lead time is constant
and the demand is constant
and shortage is allowed
and supply rate is infinite
and increment discount
Then apply the constant model Q9

Rule 10 If the lead time is constant
and the demand is constant
and shortage is allowed
and supply rate is finite
and no discount
Then apply the constant model Q10

Rule 11 If the lead time is constant
and the demand is constant
and shortage is allowed
and supply rate is finite
and all units discount
Then apply the constant model Q11

Rule 12 If the lead time is constant
and the demand is constant
and shortage is allowed
and supply rate is finite
and increment discount
Then apply the constant model Q12

Rule 13 If the lead time is constant
and the demand is stochastic
Then apply probabilistic model (s,S)

Rule 14 If the lead time is stochastic
and the demand is constant
Then apply the probabilistic model (s,S)

Rule 15 If the lead time is stochastic
and the demand is stochastic
Then apply probabilistic model (s,S)

Rule 16 If the lead time is constant
and the demand is linear
Then apply Mitra et al model

Rule 17 If the lead time is constant
and the demand is seasonal on constant base
Then apply constant based seasonal model

Rule 18 If the lead time is constant
and the demand is seasonal on linear base
Then apply the linear based seasonal model

Rule 19 If the lead time is constant or stochastic
and the demand is slow-moving
Then apply the model (s, s+1)

6.6.2 Rules for Setting Reorder Point

Rule 20 If the lead time is constant
and the lead time is less than time supply
and the demand is constant
Then set reorder point using equation (5-41)

Rule 21 If the lead time is constant
and the lead time is greater than time supply
and the demand is constant
Then set reorder point using equation (5-43)

Rule 22 If the lead time is constant
and the demand follows a Normal distribution
Then set reorder point using equation (5-44)

Rule 23 If the lead time is constant
and the demand follows a Poisson distribution
Then set reorder point with equation (5-45)

Rule 24 If the lead time is constant
and the demand follows a empirical distribution
Then set reorder point using equation (5-46)

Rule 25 If the demand is constant
and the lead time follows a Normal distribution
Then set reorder point using equation (5-47)

Rule 26 If the demand is constant
and the lead time follows a Poisson distribution
Then set reorder point using equation (5-48)

Rule 27 If the demand is constant
and the lead time follows a empirical distribution
Then set reorder point using equation (5-50)

Rule 28 If the demand is probabilistic
and the lead time is probabilistic
Then set reorder point using equation (5-52)

Rule 29 If the demand is slow-moving
and the lead time is constant
Then set reorder point using equation (5-54)

Rule 30 If the demand is slow-moving
and the lead time follows a Normal distribution
Then set reorder point using equation (5-47)

Rule 31 If the demand is slow-moving
and the lead time follows a Poisson distribution
Then set reorder point using equation (5-48)

Rule 32 If the demand is slow-moving
and the lead time follows a empirical distribution
Then set reorder point using equation (5-50).

6.7 Development of the Help System

A help system provides the user with on-line information about how to operate the intelligent inventory decision support system. The creation of a help system for a windows-based application compromises the following major tasks [163]:

- Planning the help system.
- Writing the text for the help topics.
- Programming the help system.
- Testing and debugging the help system.

To the user, the help system appears to be part of the application, and is made up of text and graphics displayed in the help window in front of the application when the user asks for help.

6.7.1 Design of the Help System

The initial task for developing the help system is to work out a plan for creating the system. It covers the following aspects:

6.7.1.1 Selecting the Contents

After the information used for creating the help system has been gathered, we have to decide what topics should be included in the help system. There are no commonly accepted rules for determining the overall content of the help system [163]. Novice users need help in learning commands, definitions and how to carry out procedures.

More sophisticated users occasionally seek help with procedures or terms, but most often need to refresh their memories with details of the of commands and functions. Therefore, the efficient help system should include the topics that are numerous enough and specific enough to provide the user with the help they need. The proposed help system is designed to provide help for all types of users, it documents all commands, procedures, definitions, and other relevant aspects of the knowledge-based inventory system.

6.7.1.2 Choosing the Structure

The help topics are organised hierarchically (Fig. 6.16) in the help system. At the top level of the hierarchy is an index which lists individual categories of topics available to the user. Moving from the index to a topic, the user goes from the general to the specific. Each successive step takes the user one level down in the hierarchy of the help system until the user reaches topic information. The hierarchical relationship of help topics determines how the user navigates through the help system.

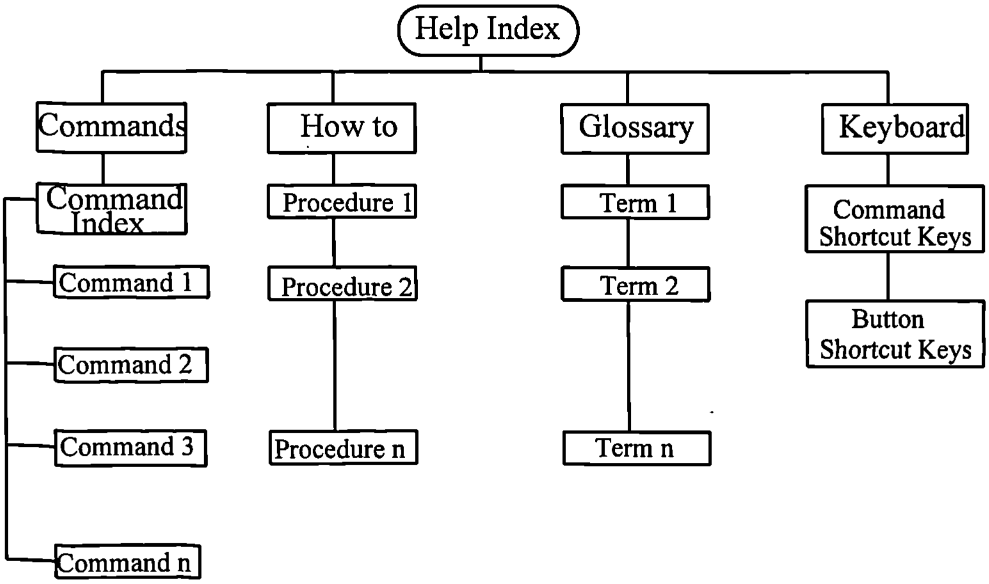


Figure 6.16 The Structure of the Help System

The help index illustrated in Fig. 6.16 lists four categories of topics. The "commands" category contains the help information about all the commands listed in the menus. The "how to" category includes the step by step instructions of carrying out procedures. The "glossary" category gives the definitions of the terms related to the domains of inventory and statistics. The "keyboard" category tells the user how to operate the knowledge-based inventory system with shortcut keys if the user has no mouse installed in his computer.

6.7.1.3 Determining the Topic File Structure.

The topic files contains the text for help topics. When the help system is built, these topic files are used to create the information displayed for the user in the help window. In this system, the basic structure of the topic files is to put the related topics together into a topic file and separate these topics by means of white space.

When the structure of the topic files has been chosen, the next step is to decide the scope and content of the topic files. In fact, all help topics could be placed in a single file or each help topic could be placed in separate file. If the help topics are placed in a single file, it will need longer time scale for the computer to locate the help topic that the user wants to see. However, an enormous single file can present difficulties during the programming of the help system. In this system, the help topics are classified into five topic files.

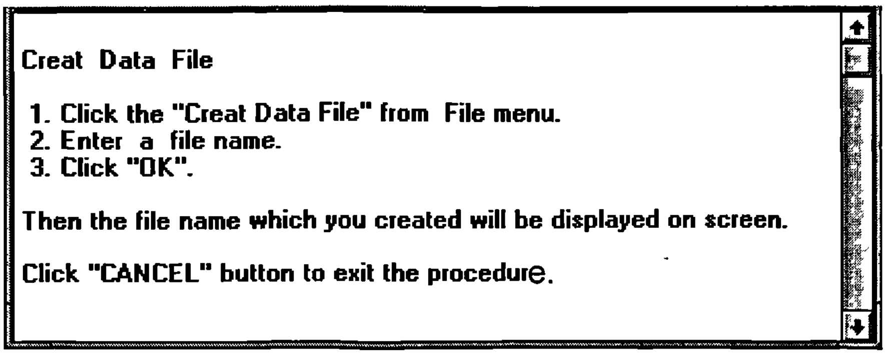
They are an index topic file and four subject topic files. The subject topic files are the command topic file containing all commands help topics, procedure topic file including all the instructions to complete procedures and keyboard topic file covering the shortcut keys of operating the system. Appendix 1 contains the details of these topic files.

6.7.1.4 Designing the Help Topics

The appearance of the help text in the help window is determined by the layout of the help topic. The visual basic supports a number of text attributes such as different fonts, point sizes, colours and graphic images to design a help system. Therefore, the help topics are not limited to plain and unformated text. Fig. 6.17 illustrates an example of the help window when the user asks for help in creating a data file.

In designing the help text appearance, the following factors have been taken into account:

- Using a minimum text and short paragraphs. Pigford and Baur [71] indicate that the reading speed decreases by 30% when users read text on a screen. Concise text will compensate for the decreased reading speed and using short paragraph can help avoiding the user becoming overloaded with the screen text.
- Using appropriate language and avoiding the unfamiliar terms and concepts.
- Using the white space to group the help text more readable.
- Using the highlighting technique to indicate context-sensitive help topics.



Creat Data File

1. Click the "Creat Data File" from File menu.
2. Enter a file name.
3. Click "OK".

Then the file name which you created will be displayed on screen.

Click "CANCEL" button to exit the procedure.

Figure 6.17 An Example of Help Topic

6.7.2 The Basic Functions of the Help System

The help menu is arranged in the menu bar of the application window (Fig. 6.18). By selecting the "Help index" command, the user can view the contents of the help information. The "How to use help" command displays the information about operating the help system. The last command is the declaration of the system.

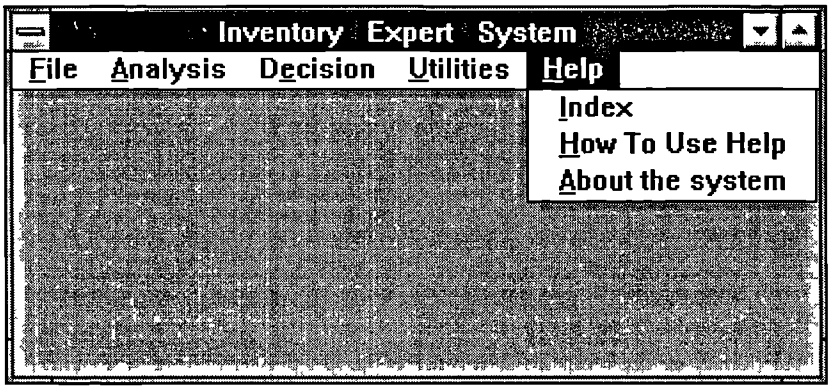


Figure 6.18 The Help Menu

To get appropriate help information, the user needs to choose the "Help index" command to open the help window (Fig. 6.19). From the menu bar of the help window, the user can select one of the following commands.

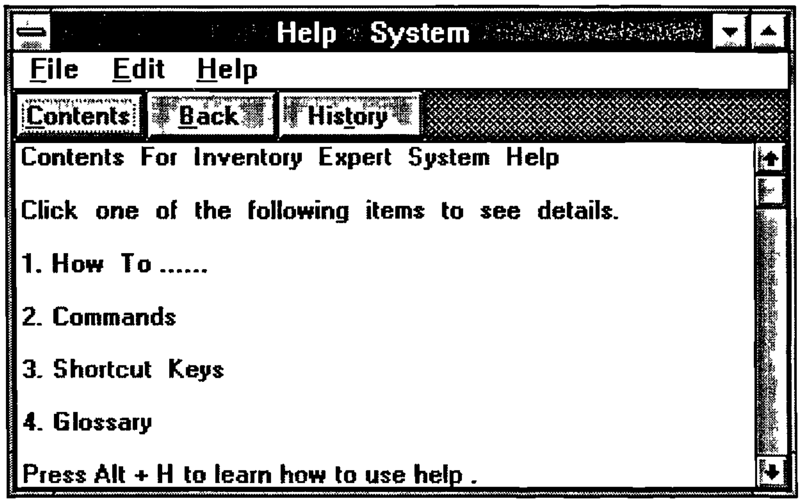
- Printing the help topic. Instead of reading the help text on the screen, the user can read help text on paper using the printing function to send the help topic to the printer.

- Annotating the help topics. When the user becomes familiar with the application. The annotating function allows the user to add more information to the correspondent help topic if the user feels it is not detailed enough.

- Copying the help topic to the clipboard. If the help topic is copied to the clipboard, it can be pasted into any kind of word processor, e.g. Microsoft Write, Microsoft Words, WordPerfect 6.0, etc. Therefore it is very easy to make manual by means of copying help topic function.

- Displaying the information about how to use the help system.

- Exiting the help system



Contents For Inventory Expert System Help

Click one of the following items to see details.

1. How To ......

2. Commands

3. Shortcut Keys

4. Glossary

Press Alt + H to learn how to use help .

Figure 6.19 The Help Window

The user can also choose one of the buttons beneath the menu bar of the help window.

Content Listing the index of help topic files. The items of the index are context-sensitive. When the user clicks any highlighted index item, the help system will access to one of the four topic files. Then the user can navigate through the topic file to find the help text.

Back returning to retrace the path through the help topics that the user has already viewed until reach help index.

History Listing a sequential list of every help topic that the user has accessed during the current session. The hierarchical structure constrains the user to navigate up and down the help topics. This function provides the user an alternative to jump from one topic to another, moving across categories of topics.

6.8 Summary

Following the proposed design of this system discussed in Chapter 3, the starting point for the development of the system is to design the application form or main menu (see Fig. 6.4) which controls the execution of the data manager, pattern identifier, order decision, and help facilities. The system was developed as several self-contained executable modules. The forms and modules included in the project of the system described in this thesis are listed in Table 6.3 and Table 6.4 respectively. Appendix 1 contains a brief user guide for the system.

During the development of the system, great effort (such as integrating the pattern identifier and enriching the model base) has been made to minimise the user's effort in order for the system to reach optimal decision. However, to run the system the user needs to enter the following information:

1. The user has to set up the significance levels for the testing methods adopted to identify demand patterns (see Chapter 4). Although the system itself suggests a set of default values for the significance levels, the inventory managers can use their statistical expertise in achieving a more efficient use of the system by entering desired values of significant levels instead of accepting the system's default values.

2. For the statistically unpredictable inventory items, the user should suggest a approximate distributions in according to the demand histograms given by the system.

3. The user also needs to enter the values of the supply rate (if the replenishment is in a certain rate), discount prices and break points (if there is any), and shortage cost (if shortage is allowed) for the system to choose right decision model and calculate reorder quantity.

4. Finally, the user has to supply the time zones representing demand data and lead time if they are in different units.

Table 6.3 Forms Developed in the Inventory IDSS

Number of the forms	Contents
Form1	The starting window
Form2	The central control window
Form3	The deleting file window
Form4	The renaming file window
Form5	The help system window
Form6	The inputting item's basic information window
Form7	The update item's basic information window
Form8	The deleting item window
Form9	The displaying item's basic information window
Form10	The annotating help topics window
Form11, Form16	The entering demand data windows
Form12	The entering password window
Form14, Form15	The updating demand data windows
Form17	The analysing demand pattern window (single item)
Form18	The analysing lead time pattern window (single item)
Form19	The making order window
Form20	The making order window for linear demand
Form21, Form23	The updating lead time data windows
Form22	The window for reporting analysis results
Form24	The making order window for stochastic demand
Form25	The analysing lead time pattern window (all items)
Form26, Form27	The windows for entering lead time data
Form28	The making order window for stochastic demand
Form29	The making order window for seasonal demand
Form30	The window for displaying item's lead time data
Form31	The window for reporting order quantity
Form32	The window for displaying item's demand data
Form33	The window for receiving goods
Form34	The window for issuing goods
Form36	The window listing al l the items needed replenishing
Form38	The window for recovering a deleted file.
Form39	The system initiation window
Form40	The analysing demand pattern window (all items)
Form41	The window for loading data from other applications
Form43	The window for reporting the proportions of patterns

Table 6.4 The Summary of Modules

General Module	Sub-module	Procedure	Purpose
Data-management	File-management	creatfile	Create a data file
		rename	Rename a data file
		erasefile	Delete a data file
		recover	Recover a deleted file
		merge	Merge data files
	Input-data	inputbasic	Input basic data of items
		demanddata	Input demand data
		leadtime	Input lead time data
		loaddata	Load data from other applications
	Update-data	modifybasic	Update basic information
		modifydemand	Update demand data
		modifyleadtime	Update lead time data
		alymeger	Update the analysis result
		goodsin	Receive goods
		goodsout	Issue goods
	Display-data	displaybasic	Display basic information
		displaydemand	Display demand data
		dispalyleadtime	Display lead time data
	Print-data	printbasic	Print basic information
		printdemand	Print demand data
		printleadtime	Print lead time data
	Delete-item	deleteitem	Delete an item from all data files
	System-initiation	systeminitial	Initiate the system
Analysis	Analysis-demand	analysiscontrol	Control the analysis process
		refreshdata	Read demand data
		seasontest	Test seasonality
		trendtest	Test linear trend
		randomtest	Test Randomness
		figoodness	Fit probabilistic distribution
		saveresult	Save analysis result
	Analysis-leadtim	statisticanalysis	Identify lead time patterns
		subjectestimation	Estimate lead time patterns
Decision	Order-decision	constantorder	Make order for constant demand
		proborder	Make order for stochastic demand
		seasonorder	Make order for seasonal demand

Table 6.4 The Summary of Modules (Continued)

General Module	Sub-module	Procedure	Purpose
		trendorder	Make order for linear demand
	Order-list	orderlist	List all items need reordering
Utilies		calculator	Call calculator
		calendar	Call calendar
		cardfile	Call cardfile
		excel	Call Microsoft excel
		word	Call Microsoft Word
	Conut-patters	countmodel	Compute different patterns
	Patiate-data	datapart	Partiate data file
Global definition		Globalbas	Declare the global variables
Help system		helpindex	Start help system
		usehelp	Display how to use help
		about	Brief description of the system
		helpcontent	Display contents of help system
		helpback	Go back to previous help topic
		helphistory	List help viewed topics
		printhelp	Print the current help topic
		copyhelp	Copy the current help topic
		annotatehelp	annotate current help topics

Chapter 7

SYSTEM TESTING AND DISCUSSION OF RESULTS

7.1 System Testing

The development of software systems involves a series of production activities where opportunities for injection errors due to of human fallibility are enormous. Errors may begin to occur at the very inception of the process, as well as at the later stages of design and development. Because of human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity [164].

Software testing is a critical element of software quality assurance (SQA) and represents the ultimate review of specification, design, and coding. It is a broad topic that is often referred to as verification and validation (V&V). Verification refers to the set of activities which ensure that the software correctly implements a specific function. Validation refers to a different set of activities that enable the software that has been built is traceable to customer requirements. In other word, verification means that "are we building the product right?" and validation means " are we building the right product" [165].

Considering the process from a procedural point of view, testing within the context of software engineering can be seen as a series of three steps that are implemented sequentially [166]. Initially, tests focus on each module individually (unit testing), ensuring that it functions properly as a unit. Next, modules must be assembled or integrated to form the complete system. Integration testing addresses the issues associated with the problems of system construction and interaction among individual modules. Finally, validation testing provides the final assurance that the system meets all functional, behavioural, and performance requirements.

The system verification was carried out using the unit testing while the system was developed and integration testing after the system programming was completed. Usually there are two kinds of tests to validate a system. For a system developed specially for a single client (bespoke system), the alpha testing process continues until the system developer and the system procurer reach agreement that the delivered system is representation of the system requirements. When a system is to be marketed as a software product, a testing process called beta testing is often used. Beta testing is conducted by a number of potential customers who agree to report problems to the system developers to modify the system for further beta testing or for general sale. The system validation is outside the scope of this study.

Test cases are designed to uncover errors in both unit and integration testing. A good test case is one that has a high probability of finding an undiscovered error. It is important to note that the execution of the test case is not only to ensure that all statements in the program have been executed at least once, but also to guarantee that the program is error-free at most situations after its verification (see Section 7.1.3). In the subsequent discussion only test samples are given to demonstrate the nature of system testing and avoiding unnecessary repetition.

7.1.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of a software design, i.e. the module. Using the detailed design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The procedure of unit testing is illustrated in Fig. 7.1. The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. All independent paths (basis paths) through the control structure are exercised to ensure that all statements in a module have been executed at least once. Finally, all error-handling paths are tested.

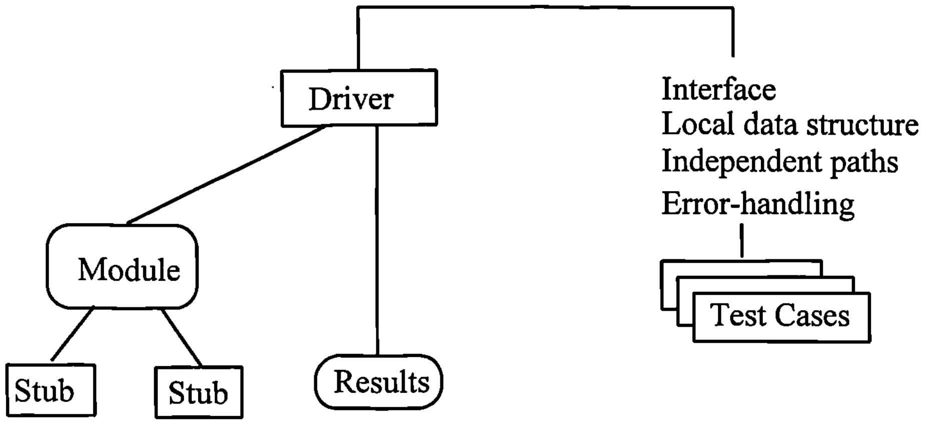


Figure 7.1 Illustration of Unit Testing

Because a module is not a stand-alone program driver and/or stub must be developed for each unit test. A driver is basically a "main program" that accepts test case data, passes them to the module (to be tested), and displays the results. A stub serves to replace subroutines called by the module to be tested. It may do minimal data manipulation, displays verification of entry,, and then returns control to the module.

7.1.1.1 Module Interface Testing

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are meaningless. The checklist for module interface test proposed by Myers [167] emphasises the following aspects:

1. the number of input parameters must be equal to the number of arguments.
2. the parameters and arguments attributes must be matched.
3. the file attributes must match the OPEN/CLOSE statements.

4. the end-of-file conditions I/O errors have to be handled.

The testing cases should be able to uncover errors caused by passing improper values to a module. For example, the following code calculates factorial *n (n!)*:

```
Function factorial (n As Integer) As  Integer
Dim result As Integer
results = 1
For i% = 1 To n
        results = results * i%
Next i%
factorial = results
End Function
```

This program is workable, however, if a number assigned to *n* is greater than 8, an overflow error will occur because the value is not within the integer range. The maximum values that can be accepted by the program to carry out factorial n within Visual Basic have been evaluated and listed in Table 7.1. The above code also has no error message that will alert users when an error has occurred. Therefore, the program can be changed as follows, in order to ensure it is error-free although the running time will be slightly longer.

```
Function factorial (n As Integer) As  Variant
Dim result As double
results = 1
If  n > 171 Then
        MsgBox "The number is too big", 48, "Error   Warning   Information"
        Exit Function
Else
For i% = 1 To n
        results = results * i%
Next i%
factorial = results
End If
End Function
```

Table 7.1 The Maximum Values of *n* for Calculating *n!* in Different Data Types

Type Name	Description	Range	Maximum Value of *n*
Integer	2-byte integer	-32,768 to 32,767	8
Long	4-byte integer	-2,147,483,648 to 2,147,483,647	13
Single	4-byte floating-point number	-3.4028E38 to 3.428E38	35
Double	8-byte floating point number	-1.7976D308 to 1.7976D308	171
Variant	can store any type of data		171

7.1.1.2 Local Data Structure Testing

The local data structure for a module is a common source of errors. Test cases should be designed to uncover errors in the following categories[144]:

1. Erroneous initialisation or default values.
2. Misspelt or truncated variable names.
3. Inconsistent data types.

The impact of global variables on a module should be ascertained during unit testing. For example, the following code is used to calculate the average demand.

```
Function average (n As Integer, D(i%) As double ) As  Double
ReDim mean As Double
For i% = 1 To n
        mean = mean + d(i%) / n
Next i%
average = mean
End Function
```

If the variable *mean* was declared in other modules and has been referenced, its initial value may be non-zero which will give an incorrect result. In order to avoid this kind of error, initialisation statement should be added into this program, i.e.

```
Function average (n As Integer, D(i%) As Double ) As  Double
```

```
ReDim mean As Double
mean = 0
For i% = 1 To n
        mean = mean + d(i%) / n
Next i%
average = mean
End Function
```

7.1.1.3 Independent Paths Testing

Testing the execution paths of a software module is an essential task during the unit test. Test cases should be designed to uncover errors due to erroneous computation, incorrect comparisons, and improper control flow.

The more common error in computation is misunderstanding and incorrect arithmetic precedence. At the very beginning, the following program is used to find out the constant demand pattern.

```
Function Constant (n As Integer, D(i%) As Double ) As String
If average(n, D(i%)) = D(1) Then
  constant = "Yes"
Else
  constant = "No"
End If
End Function
```

The basic idea of this program is that if all the demand data have the same value, the average should equal to one of the demand data. However, it will give a wrong result if the demand data are:

1, 1, 1, 0, 2, 0, 0, 1, 1, 2, 2, 1.

Apparently, the demand is not constant although both the average and *D(1)* are equal to 1. To overcome the defect of this program, we need to count the number (stored in number%) of the demand data which have the same value. If the number is equal to the total number (*n*) of demand data, the demand is constant. The following is the modified program.

```
Function Constant (n As Integer, D(i%) As Double ) As  String
number% = 0
For i% = 1 To n
   If D(1) = D(i%) Then   number% = number% + 1
Next i%
If number% = n Then
   constant ="Yes"
Else
   constant ="No"
End If
End Function
```

Another important unit testing task is to exercise loops which are the cornerstone for the vast majority of all algorithms implemented in a software. During loop testing, *n, n-1, n+1* passes through the loop should be applied because a loop often fails at its boundaries, where *n* is the maximum number of allowable passes through the loop and EOF (end-of-file) stands for the end of a file. The following program was designed to read data from two files.

```
Sub Read-data
file1% = FreeFile
file2r% = FreeFile
Open "f:\ltd\vb1\status.inf" For Input As file1%
Open "f:\ltd\vb1\item.inf" For Input As file2%
Do Until EOF(file1%)
    Input #file1%, item1$, item2$, item3!, item4!, item5!, item6%, item7$
    Input #file2%, tem1$, tem2$, dat1!, dat2!, dat3!, dat4!, dat5$
    If item1$ = tem$ Then
         Close
         Exit Sub
    End If
Loop
End Sub
```

Late testing found out that the program will fail if the two files contain different numbers of recorders. Therefore, the correct program should be as follows:

```
Sub Read-data
file1% = FreeFile
file2r% = FreeFile
Open "f:\ltd\vb1\status.inf" For Input As file1%
Open "f:\ltd\vb1\item.inf" For Input As file2%
Do Until EOF(file1%)
  If EOF(file2%) Then
      MsgBox "End of File.", 64, "Read Data"
    Close
    Exit Sub
  Else
    Input #file1%, item1$, item2$, item3!, item4!, item5!, item6%, item7$
    Input #file2%, tem1$, tem2$, dat1!, dat2!, dat3!, dat4!, dat5$
    If item1$ = tem$ Then
        Close
        Exit Sub
    End If
  End If
Loop
End Sub
```

7.1.1.4 Error-Handling Testing

Error-handling testing is the last step of unit testing. An error-handling path is used to reroute or cleanly terminate processing when an error occur. Instead of reporting the error message, sometimes the error-handling paths should have the ability to remove erroneous results which may affect further performance. For example, the Visual Basic does not have a function to delete a recorder from a data file. An alternative is to create a temporary file and copy all the recorders except the one you want to delete from the original file to temporary file. Then delete the original file and rename the temporary file as the original file. During this procedure an error may occur to diverse the processing to error-handling path. If the error-handling path only reports the error message without erasing the temporary file, you may get a wrong result when you want to delete another recorder because there may be some recorders remained in the temporary file when the last error occurred.

7.1.2 Integration Testing

When the error-free individual modules are put together to form a system, it may not works properly. Data can be lost across modules; one module can have an inadvertent adverse effect on another; subfunctions, when combined, may not produce the desired major function; individually acceptable imprecision may be magnified to unacceptable levels; global data structures can cause problems. The integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with inter-modules.

The most common methods of integration testing are top-down integration and bottom-up integration [145]. Top-down integration is an increment approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control module. Modules subordinate to the main control module are incorporated into the system structure. Bottom-up integration begins constructing and testing with atomic modules (modules at the lowest level in the system structure), and then moves upward to the top level. Because modules are integrated from bottom up, testing required for modules subordinate to a given level is always available. This approach has been adopted for the integration testing of the system described in this thesis.

7.1.3 The Selection of Test Cases

To ensure the quality of the system, the test cases must have the highest likelihood of finding the most errors with a minimum amount of effort. The following points are important to observe while constructing testing cases.

1. The test cases must cover a wide range of samples to verify algorithms at different conditions. For example, the variability coefficient (VC) [36]

$$vc = \frac{\frac{1}{N}\sum_{j=1}^{N}[D(j)]^2 - [E(D)]^2}{[E(D)]^2} = \frac{N\sum_{j=1}^{N}[D(j)]^2}{[\sum_{j=1}^{N} D(j)]^2} - 1$$

was used to identify randomness of demand data. It can give satisfactory results in most cases. However, if an item's annual demand data are

1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0.

Apparently, this item has a low demand pattern. However, the VC value of this item is 5 which suggests the demand is random. The above example indicates that this method is not suitable to test the items with low average and zero demand in some intervals. Therefore, the VC method was replaced by the turning point test to identify the randomness of an item's demand data.

2. The test cases should include extreme samples because an algorithm may fail at these conditions. For example, the rank test was used to identify the trend of demand data. The rank test statistic described in [121] will suggest a trend if the demand data are

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2.

The problem is that the test statistic can only be used when there are no ties in the sample data (not mentioned in [121]). If there are some ties in the data series (in the example all the data have the same value), the statistic must be modified (see Section 4.3.2).

3. The test cases must be correct to avoid confusing the test results. During the integration testing, a two year demand data of more than 10,000 spare parts from one of the co-operating companies has been loaded into the system. When testing the function of "setting maximum inventory level", an error of "divided by zero" occurred. It was found that it was not the program itself but the incompleteness of original data caused this error.

7.2 Sources of Testing Data

In order to base the current research on real inventory situations, collaboration with industrial organisation was sought. At the onset of the study, links were established with Optical Fibres, a high tech organisation which produces optical fibres and is located at the industrial park of Deeside, UK. Many visits were made to Optical Fibres to understand the inventory management system and to collect the relevant data. At a later stage of this research, links were established with British Airways which also provided inventory data. In what follows we discuss the inventory management system of Optical Fibres and the data provided by both of the co-operating companies.

7.2.1 Optical Fibres

The company has five divisions (Fig. 7.2), and is the largest fibre manufacturer in Europe and the third largest in the world. The material department under the manufacturing division deals with a total of 8,425 items with value of approximately 3 million pounds (see Table 7.2).

Table 7.2 Summary of Inventory in Optical Fibres

Group	Item	Value(£)
Process Materials	6220	1,023,507.25
Spare Parts	2070	2,159,676.90
Consumable	135	52,051.79
Total	8425	3,235,235.94

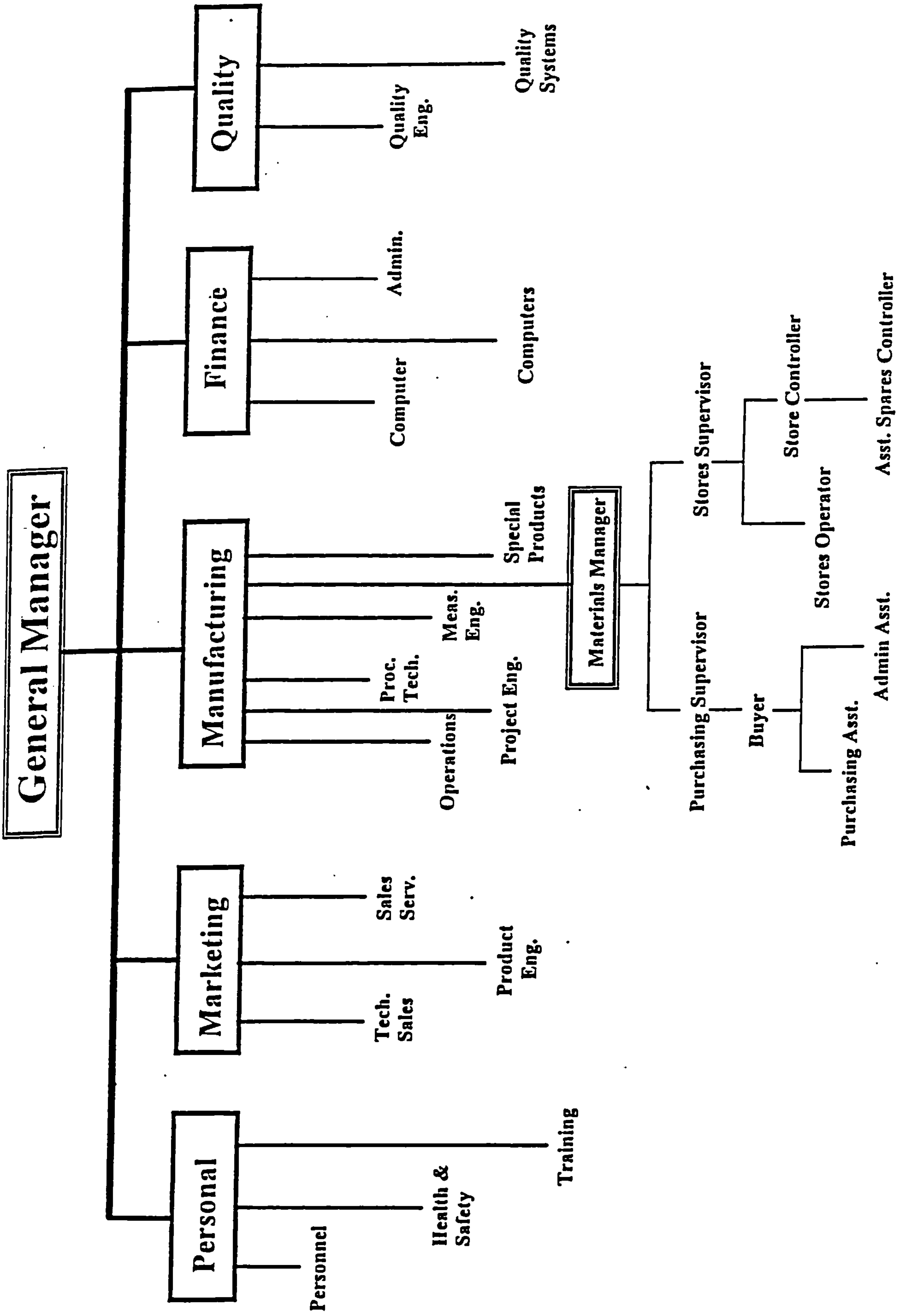


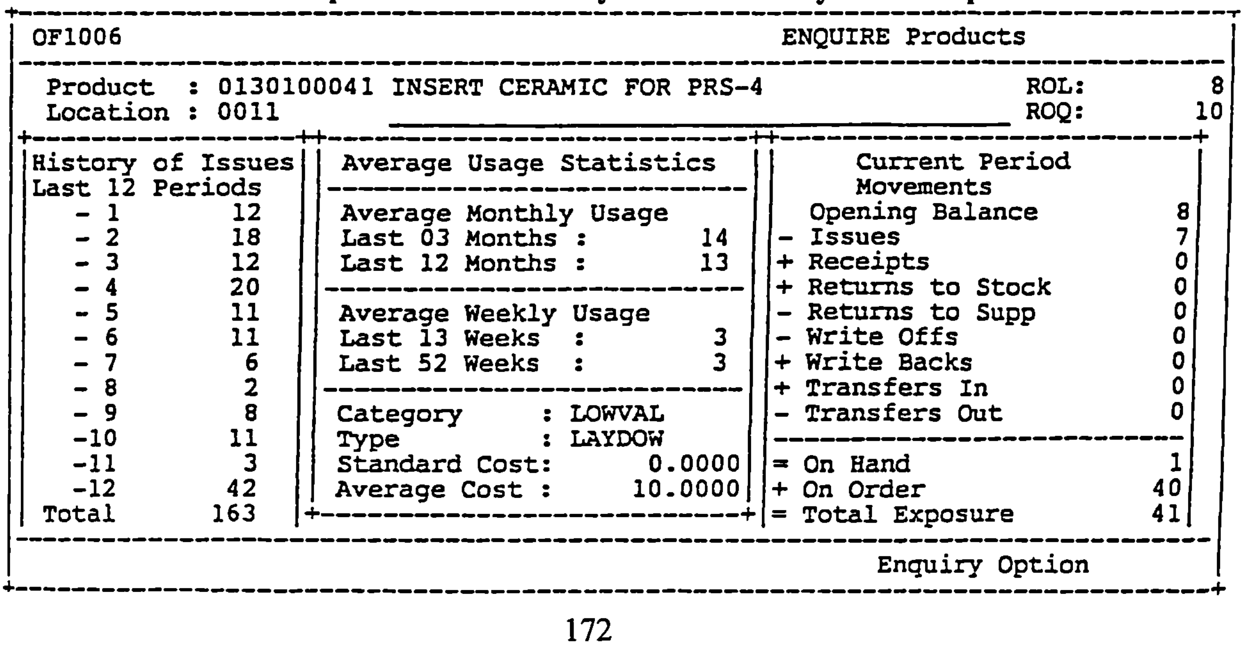
Figure 7.2 The Organisation of Optical Fibres

For controlling the stock, a maximum level and a minimum level have been established for each item. The minimum level which is similar to the order point was estimated by the technicians at the very first time when the company was established, and the maximum level is equal to the sum of the minimum level and the order quantity. Although there is a maximum level in the system, unfortunately, this parameter has never been applied.

New orders are triggered by two events: (1) special requisition and (2) purchasing report. The special requisition is submitted by different divisions for ordering casually used materials, and the purchasing report is daily generated by the inventory controller using computers. The order quantities are decided based on the following five factors: restrictions of supplier and transportation, average usage, current storage, lead time, and buyer's experience. The unit price of materials is negotiated once a year and the kind of currency used in paying is unchangeable once it is decided. The future demands of materials are forecasted using the moving average of the last 12 months/52 weeks.

The current computer-based inventory management system used in Optical Fibres can supply three categories of information (see Table 7.3): history of issues; average usage statistics; and current period movements. Apparently, the system can only be used to monitor the inventory status and lacks the ability to suggest a suitable inventory model for making order decisions.

Table 7.3 Sample Results Given by the Current System of Optical Fibres



```
+------------------------------------------------------------------------------+
| OF1006                                            ENQUIRE Products           |
|------------------------------------------------------------------------------|
|  Product  : 0130100041 INSERT CERAMIC FOR PRS-4                  ROL:       8|
|  Location : 0011       ______________________________________    ROQ:      10|
+------------------++-------------------------++-----------------------------+
|History of Issues || Average Usage Statistics ||     Current Period         |
|Last 12 Periods   ||-------------------------||     Movements              |
|  - 1        12   || Average Monthly Usage    ||   Opening Balance        8 |
|  - 2        18   || Last 03 Months :      14 || - Issues                 7 |
|  - 3        12   || Last 12 Months :      13 || + Receipts               0 |
|  - 4        20   ||-------------------------|| + Returns to Stock       0 |
|  - 5        11   || Average Weekly Usage     || - Returns to Supp        0 |
|  - 6        11   || Last 13 Weeks  :       3 || - Write Offs             0 |
|  - 7         6   || Last 52 Weeks  :       3 || + Write Backs            0 |
|  - 8         2   ||-------------------------|| + Transfers In           0 |
|  - 9         8   || Category      : LOWVAL   || - Transfers Out          0 |
|  -10        11   || Type          : LAYDOW   ||---------------------------|
|  -11         3   || Standard Cost:    0.0000 || = On Hand                1 |
|  -12        42   || Average Cost :   10.0000 || + On Order              40 |
|  Total     163   |+-------------------------+|| = Total Exposure        41 |
|------------------------------------------------------------------------------|
|                                                  Enquiry Option              |
+------------------------------------------------------------------------------+
```

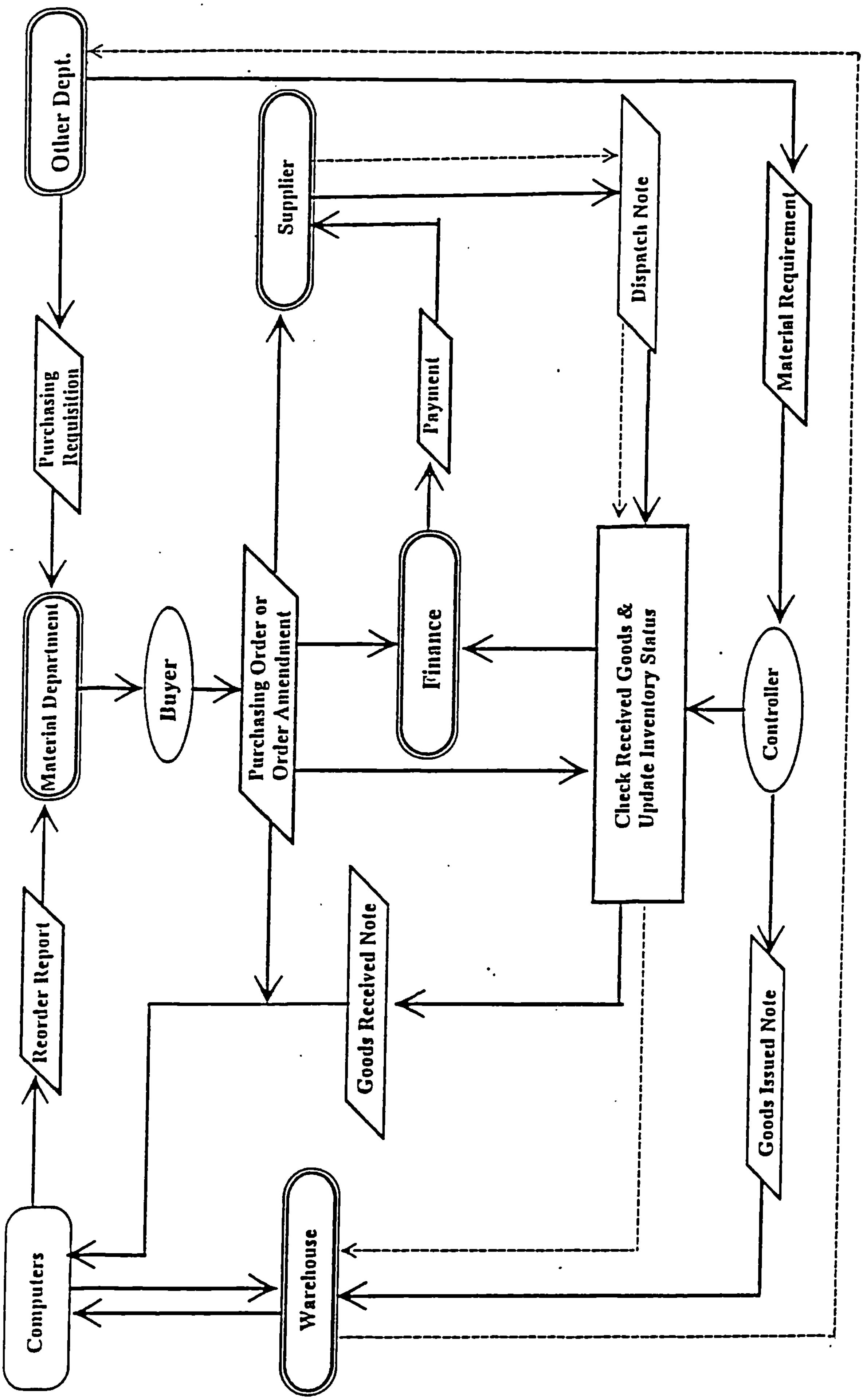


Figure 7.3 The Process of the Present Inventory Management System in Optical Fibres

Fig. 7.3 illustrates the process of the current inventory management system in Optical Fibres. The buyers are responsible for: 1) estimating usage; 2) estimating the lead time; 3) negotiating unit price; 4) choosing suppliers; 5) making orders; and 6) presenting analysis reports. The duties of the inventory controllers are: 1) examining delivered materials and filling in the Goods Received Note; 2) issuing orders for required materials and filling in the Stores Requisition; 3) inputting both Goods Received Note and Stores Requisition into computer system; 4) presenting reorder report; 5) presenting material consuming report; and 6) analysing the maintenance plans.

As more competitors are appearing in this field, and the prices of the products have dropped rapidly (from 50p per metre in 1980s to 4.6p per metre in 1990s), at present the company is facing the challenge to reduces product costs. This is closely related to the effectiveness of inventory management which can reduce both the storage and the labour operating expenses, as a small percentage reduction in inventory will be transformed into large operating profit.

Most of the inventory models were developed to minimise inventory costs based on the assumption that these costs were known beforehand. In fact, companies usually do not keep the records of inventory costs because they are very difficult to be obtain accurately. The following costs information of Optical Fibres was estimated by the purchasing supervisor of the material department.

• **Holding costs:**	**100%**
Labour	47.23
Insurance	1.55
Accommodation	11.31
Interest(6%)	39.02
Depreciation	0.89
• **Purchasing costs:**	**100%**
Labour	69.06
Accommodation	3.99
Communication & Computer	5.99
Transportation	16.97
Stationery	3.99

Table 7.4 Sample Data from British Airways

Part Number	Nov94	Dec94	Jan95	Feb95	Mar95	Apr95	May95	Jun95	Jul95	Aug95	Sep95	Oct95	Last Year total	Previous Year total	Price £	Lead Time (day)
19	6	11	20	0	3	22	4	4	3	8	4	7	98	53	130.20	112
20	10	7	5	11	15	19	20	5	2	1	14	18	83	53	19.51	114
35	5	10	13	7	4	13	6	3	0	11	3	5	75	85	20.92	82
44	0	9	7	1	10	3	1	3	5	3	14	1	86	86	109.62	110
58	10	6	3	5	2	1	1	2	3	13	0	3	53	33	11.62	123
66	4	2	0	10	2	2	2	4	12	8	4	2	56	29	118.00	93
67	8	10	1	11	2	3	0	4	2	3	1	3	63	63	206.60	114
68	7	4	9	0	10	2	9	1	0	2	3	4	44	32	25.26	123
70	2	14	8	8	9	2	6	4	2	6	4	4	67	87	17.18	123
73	3	3	8	10	12	5	9	11	6	2	4	3	49	15	9.38	83
80	11	6	7	4	10	10	9	4	2	1	7	4	74	20	39.18	84
92	1	4	2	2	3	3	1	2	12	2	0	4	17	10	85.88	110
93	6	2	6	0	2	11	0	4	5	4	3	6	37	44	11.36	123
94	2	2	7	3	7	2	10	0	0	2	2	4	29	36	25.50	93
121	6	9	4	2	2	1	1	1	1	2	0	4	45	32	10.84	123
127	1	1	2	0	2	2	2	0	1	0	0	9	9	38	22.93	92
129	8	3	4	2	1	8	5	5	8	7	3	6	43	45	233.74	91
606	0	0	1	1	0	0	1	1	1	0	0	0	3	0	171.68	84
607	1	0	0	1	1	0	1	0	0	0	0	0	7	7	23.32	91
1709	5	3	5	5	2	2	3	3	0	2	4	10	33	28	131.22	50
1710	9	9	6	3	7	9	4	3	5	8	10	8	100	67	112.21	39
1711	4	4	4	10	2	4	5	7	2	1	3	2	31	4	144.80	59
1712	2	2	10	4	2	2	3	3	1	2	0	3	34	9	23.79	39
1805	2	1	2	3	2	3	3	1	5	2	5	2	17	22	34.98	46
1806	1	2	0	1	1	1	5	2	2	1	1	0	18	8	186.82	63
7364	11	5	6	3	6	1	2	2	4	3	5	11	70	32	4.64	58
7365	1	1	7	7	2	2	3	1	5	10	0	0	61	31	4.28	52

The holding cost of an individual item is proportional to unit price, and the purchasing cost of an individual item is proportional to the number of orders which is at the average number of 30~35 orders per day. In the system testing below, we assumed that the holding cost is 20% of unit price and the ordering cost is £25 per order.

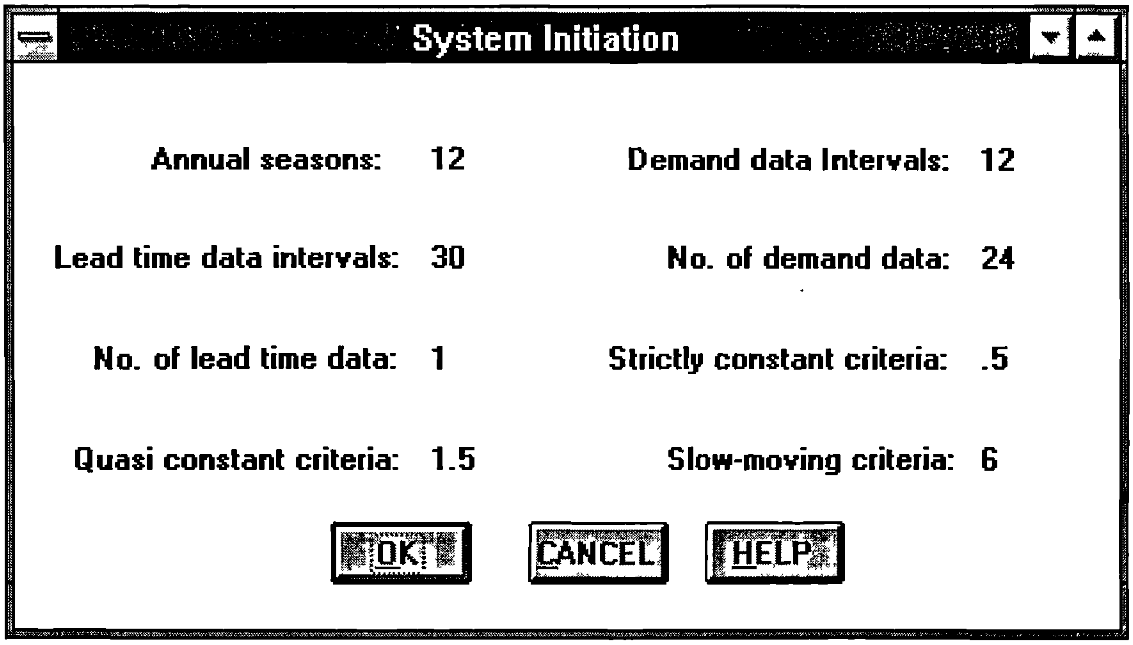
7.2.2 British Airways

At a later stage of this research, links were established with the engineering division at BA. The company was looking for modern approaches to manage their large scale inventories of the aircraft spare parts.

The number of spare part items for a single aircraft model is in the order of 5,000. Real sample data were provided and runs were conducted. The first was 5,000 items covering a 12 month period and the second was for 15,731 items covering a 24 month period. Table 7.4 shows sample data from British Airways.

7.3 System Initiation

To run the system the user needs to specify the values of parameters used to configure the system (see Fig. 7.4)


System Initiation

Annual seasons: 12

Demand data Intervals: 12

Lead time data intervals: 30

No. of demand data: 24

No. of lead time data: 1

Strictly constant criteria: .5

Quasi constant criteria: 1.5

Slow-moving criteria: 6

Figure 7.4 The System's Initiation Window

The values of these parameters specified by the user for a practical situation may affect the results of both pattern identification and order decision (see Section 7.4). Table 7.4 summarises the definitions of theses parameters and the possible values that the user can choose for each parameter.

Table 7.5 Values of Parameters for System Initiation

Parameter	Definition	Units / User's input
Annual Seasons	The length of the seasonality or the number of seasons in a year.	Monthly/12; Bimonthly/6; Quarterly/4; Semiannual/2.
Demand data Intervals	The time period strictly between successive demand data.	Weekly/52; Monthly/12; Bimonthly/6.
Lead time data intervals	The time period strictly between successive lead time data.	Daily/1; Weekly/7; Monthly/30.
No. of demand data	The total number of available demand data for analysis.	Depends on particle situation.
No. of lead time data	The total number of available lead time data for analysis.	Depends on particle situation.
Strictly constant criteria	Critical value (standard deviation) to define the strictly constant.	Specified by user.
Quasi constant criteria	Critical value (standard deviation) to define the quasi constant.	Specified by user.
Low demand criteria	Critical value (annual demand) to define the Low demand.	Specified by user.

7.4 Testing Results

As shown in Fig. 3.2 (Chapter 3), the pattern identifier works together with the data manager to carry out the demand analysis of inventory items; the order quantities are calculated using the models chosen by the model selector from the model base based on the results of demand analysis and the rule base. This section discusses the results produced by the system from the analysis of a real data set provided by the co-operating companies. For brevity and to maintain confidentiality only samples of the results will be given to verify the demand analysis methods, the inventory model base and the rule base of the proposed system.

7.4.1 Results of Demand Analysis

The following outputs of the system were produced by the statistical methods and the process of demand analysis detailed in Chapter 4.

The details of the input data file obtained from British Airways was saved in Excel format (see Table 7.4). This data file contains 15,731 records (inventory items), each record includes two years demand data (24 data fields) and other inventory information such as annual demand, unit price, and lead time. The data file was first loaded into Excel, then the system transferred this data into the system through its interface. The time spent on data transferring is depends on the speed, memory, and the type of the used PC. The following testing results were based on a PC-486 which has 16MB memory, 256K cache, and a CPU at 66MHz. Table 7.6 shows some statistics of the performance of transferring the data file provided by British Airways into the proposed system. The transferring speed is dependent on how many data fields contained in each record and whether the data loaded into the Excel was from a floppy disc or hard disc.

Table 7.6 Statistical Data of the Performance of Transferring Data into the System

Load data into Excel		**Transfer data into the system**	
From hard disc	From floppy disc	24 data fields	12 data fields
14 seconds	782 seconds	0.5 second/recorder	0.21 second/recorder

The data collected from Optical Fibres have to be input manually. Table 7.6 shows demand data of some inventory items collected from Optical Fibres. The system was designed to keep monthly demand data of three years for every item in the demand data file. The file is updated by adding new demand data into the file and shifting the oldest demand data out of the file, so that the analysis results can reflect the recent demand situations. If the analysis results show that an item's current demand pattern has changed, then a different inventory model will be automatically selected by the rule base to deal with the new situation.

Table 7.7 Demand Data from Optical Fibres

Period	Item				
	A	B	C	D	E
1	59	80	6	12	50
2	80	80	11	5	25
3	79	80	4	5	70
4	80	79	10	5	80
5	20	80	8	5	25
6	77	80	7	5	25
7	77	80	3	5	45
8	80	80	4	5	25
9	80	80	7	5	25
10	80	80	7	5	25
11	80	80	8	5	25
12	80	80	7	5	25
13	80		8	5	25
14	51		2	5	35
15	80		7	3	25
16	80		9	5	25
17	80		7	5	25
18	79		4	5	25
19			7	5	25
20			7	5	50
21			5	7	15
22			8	7	25
23			8	7	35
24			5	4	80
25			3	7	21
26			4	11	20
27			3	10	20
28			5	6	40
29			6	5	20
30			2	5	20

Fig. 7.5 to 7.9 show the graphical forms of the demand data contained in Table 7.7 and the frequency histograms of these demand data. This form is more useful in helping the inventory manager to make a judgement on what kind of demand pattern of an inventory item might be expected. For example, Fig. 7.6 shows that the graph (on the left) indicates the demand data of item B is near a level line and the frequency histogram of this item has only two bars. All these indicate that the demand pattern of this item might

be constant. Similarly, Fig. 7.7 shows the demand data of item C is random and the frequency histogram of the item has a bell-like-shape which means that the demand pattern of this item may fit a normal distribution. If the result of an item's demand analysis produced by the system based on the statistical methods described in Chapter 4 do not match its graphical form, it implies that the statistical methods may be improper for such case. From this point of view, the graphical form is also useful in helping to verify the output of the system's pattern identifier.

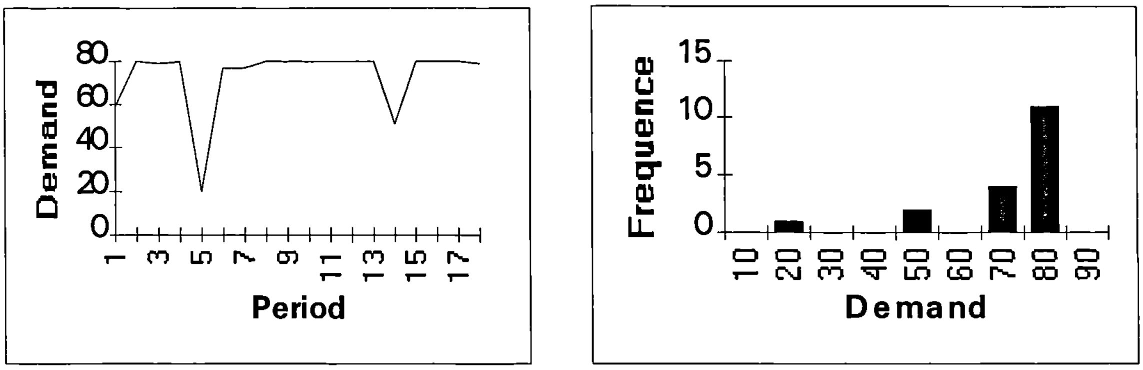


Figure 7.5 The Demand Value and Frequency Histogram of Item A

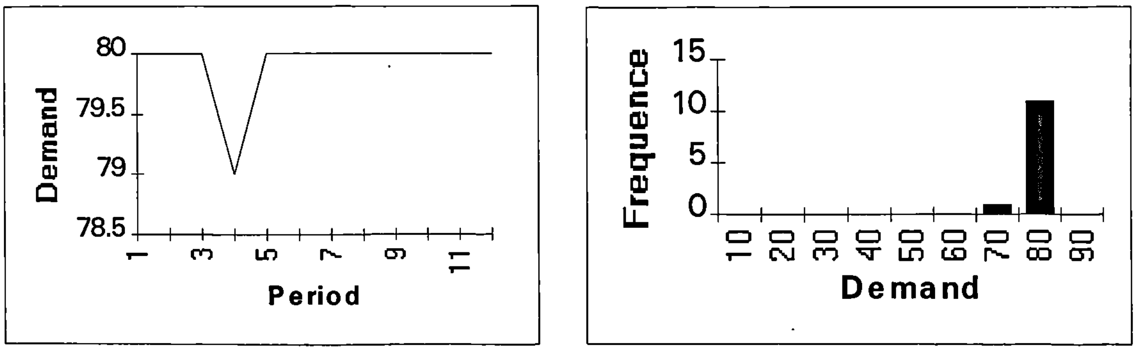


Figure 7.6 The Demand Value and Frequency Histogram of Item B

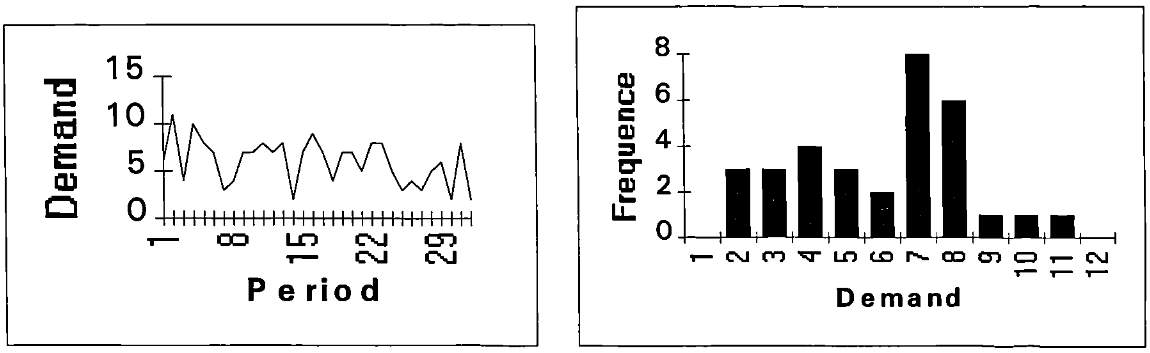


Figure 7.7 The Demand Value and Frequency Histogram of Item C

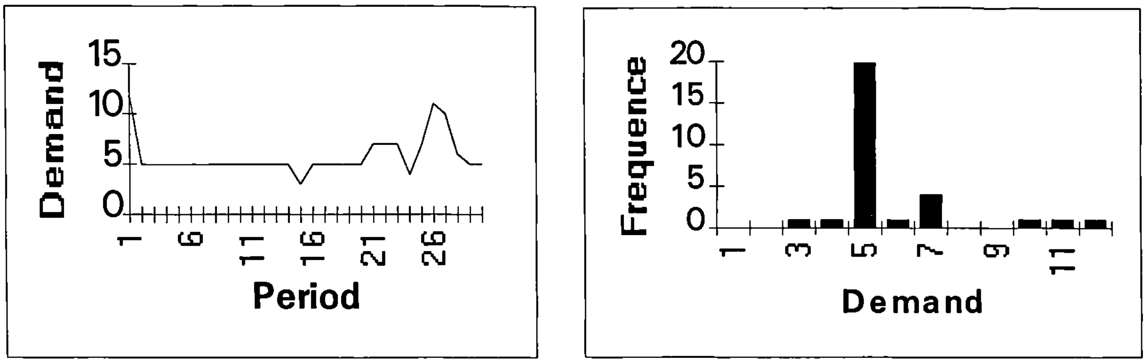


Figure 7.8 The Demand Value and Frequency Histogram of Item D

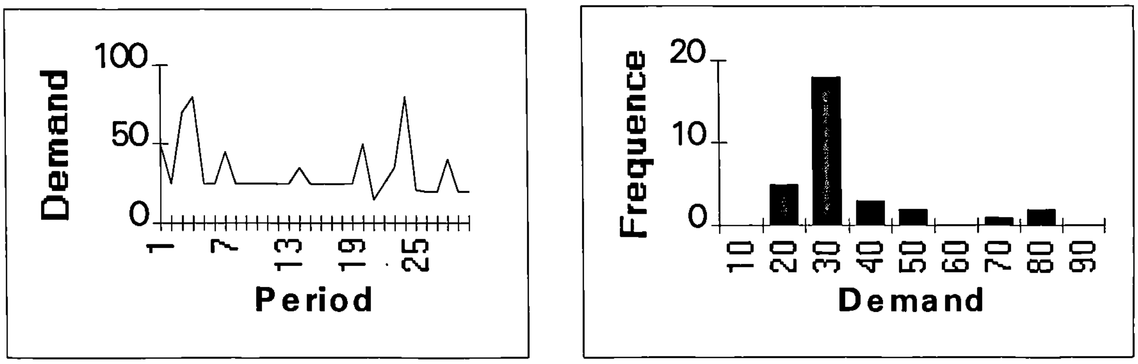


Figure 7.9 The Demand Value and Frequency Histogram of Item E

Table 7.8 shows the analysis process of the demand data given in Table 7.7. At each stage of the testing process, both the statistic values and critical values for different statistical methods are given by the system to show the user why the test hypotheses was rejected or accepted. From the user's point of view, the most important information contained in this table is the procedure of how the analysis results were achieved. The last row of each column contains the final analysis result of each item.

Table 7.8 Process of Demand Analysis (based on the data shown in Table 7.7)

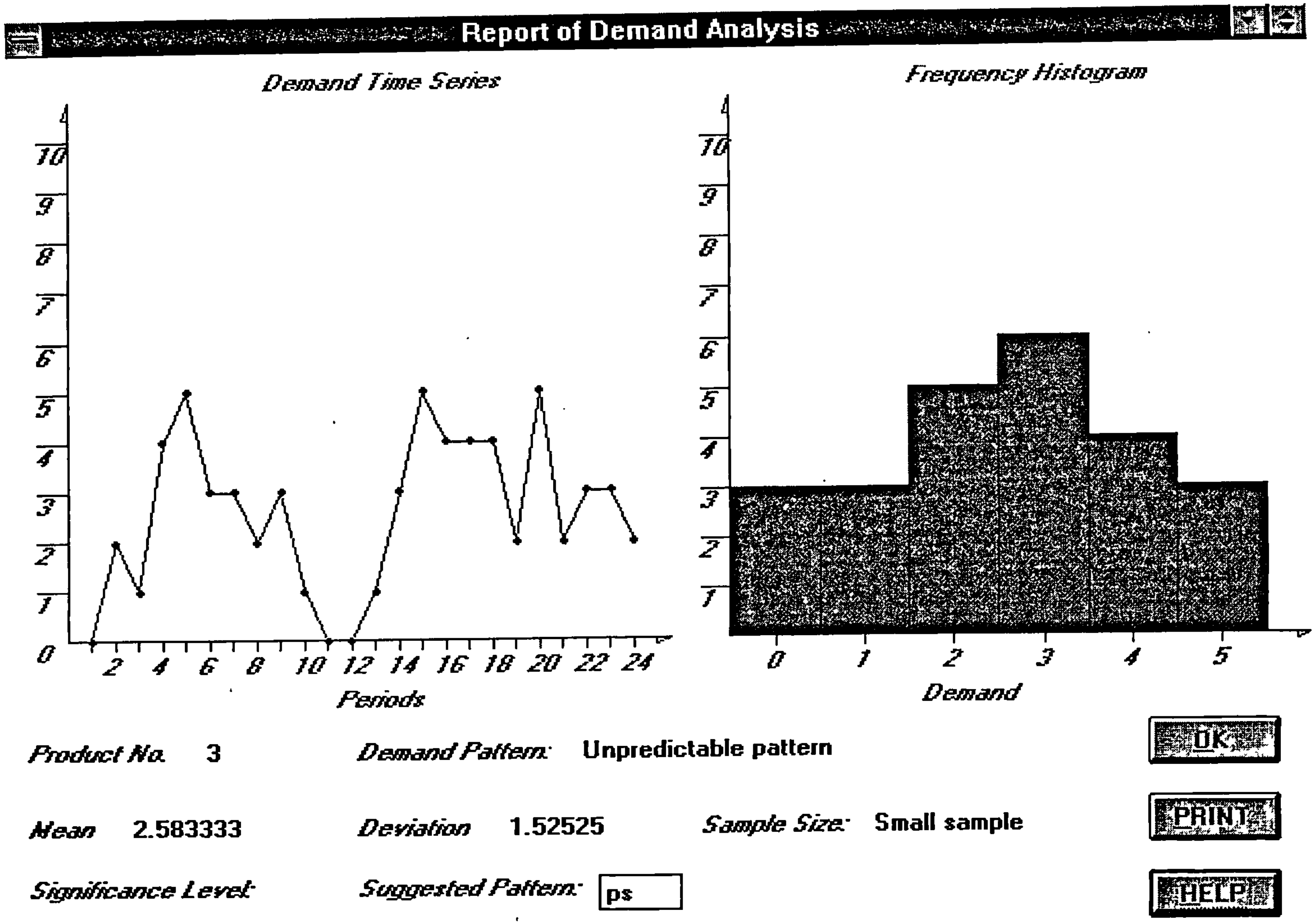
Test	Item				
	A	B	C	D	E
Seasonal	C = 7.81	C = 7.81	C = 19.68	C = 19.68	C = 19.68
	T = 2.83	T = 0.71	T = 11.0	T = 10.19	T = 10.62
	R: Reject	R: Reject	R: Reject	R: Reject	R: Reject
Trend	C = 1.34	C = 2.23	C = 1.32	C = 1.32	C = 1.32
	T = 1.06	T = 0.73	T = -0.15	T = 0.71	T = 0.69
	R: Reject	R: Reject	R: Reject	R: Reject	R: Reject
Randomness	C = 1.28	C = 1.28	C = 1.28	C = 1.28	C = 1.28
	T = 0.39	T = 2.27	T = 0.67	T = 4.87	T = 2.35
	R: Accept	R: Reject	R: Accept	R: Reject	R: Reject
Normal distribution	C = 0.29		C = 0.25		
	T = 46.94		T = 1.91		
	R: Reject		R: Reject		
Poisson distribution	C = 11.07		C = 16.92		
	T = ∞		T = 4.02		
	R: Reject		R: Accept		
Empirical distribution	R: Accept				
Strictly constant		C = 0.5		C = 0.5	C = 0.5
		T = 0.28		T = 1.61	T = 17.94
		R: Accept		R: Reject	R: Reject
Quasi Constant				C = 1.9	C = 1.9
				T = 1.61	T = 17.94
				R: Accept	R: Reject
Unpredictable pattern					R: Accept

C----The critical value; T----The value of statistic; R----The result of statistical test.

The results in Table 7.9 show the analysis outputs of BA demand data contained in Table 7.4. This form is useful in helping the inventory manager to understand the quantitative properties of inventory the items. The item's demand pattern indicates the selected model to be applied to the item. Parameters used for calculating reorder point, order quantity (such as mean, standard deviation) are also given in this table. The reorder points which decide when a new order should be placed and the maximum inventory level used to calculate order quantity for the item with low demand or probabilistic demand patterns are also given in the last two columns from the right. For items with linear or seasonal demand patterns, the decisions of when to order and how many to order are not based on the parameters of reorder point and maximum inventory level because the demand patterns of these items are time dependent (see Section 5.5).

Table 7.9 The Testing Results of the Data given in Table 7.4

Part	Demand Patter												Decision	Parameters
No.	Low	Trend		Season		Random			Constant		Parameters			
	demand	Up	Down	Constant	Trend	Normal	Poison	Empirical	Absolute	Quasi	Mean	Variance	Order Point	Max Inventory
19								yes			7.7	6.8	44	57
20								yes			10.6	6.6	56	96
35								yes			6.7	4.2	27	57
44								yes			4.8	4.4	28	39
58								yes			4.2	3.8	26	58
66								yes			4.3	3.7	21	31
67								yes			4	3.6	24	31
68						yes					4.3	3.6	26	48
70								yes			5.8	3.6	32	63
73						yes					6.2	3.5	25	70
92								yes			3	3	18	28
93								yes			4.1	3.1	24	56
94								yes			4.3	3.1	17	37
127								yes			1.7	2.4	10	24
129							yes				5	2.2	20	28
606	yes										0.42		1	2
607	yes										0.33		1	2
1709								yes			3.7	2.5	10	19
1710										yes	6.7	2.3	12	25
1711								yes			4	2.4	12	21
1712								yes			2.8	2.4	7	25
1805							yes				2.6	1.6	6	20
1806								yes			1.4	1.3	5	9
7364							yes				4.9	2.2	13	69
7365								yes			3.2	3.2	11	58
											Interception	Slope		
80			yes								9	-0.5		
121			yes								6	-0.4		



Product No. 3

Demand Pattern: Unpredictable pattern

Mean 2.583333

Deviation 1.52525

Sample Size: Small sample

Significance Level:

Suggested Pattern: ps

Figure 7.10 Result of Demand Pattern Identification

If analysis output shows that the inventory item has an unpredictable demand pattern (see Fig. 7.10), the system will ask the user to suggest a suitable pattern for the item based on the user's judgement complied with observing the demand data series and frequency histogram as plotted on Fig. 7.10. In this example, the item is specified as a Poisson distribution.

Another useful facility is that if the system cannot identify seasonal movements because of a lack of data, if the user wishes, the system can force this seasonal pattern assumption and also estimate the consequential seasonal movements.

To facilitate the user to suggest a demand pattern for an item with unpredictable demand pattern, An on-line help command button has been developed to prompt the abbreviations that the user can input to stand for different demand patterns (see Table 7.10)

Table 7.10 The Abbreviations Standing for Different Demand Patterns

Abbreviation	Standing for Demand Pattern
ss	Seasonal movements with constant base
ls	Seasonal movements with linear base
ac	Strictly constant pattern
qc	Quasi constant pattern
up	Upward linear trend
dw	Downward linear trend
nm	Normal distribution
ps	Poisson distribution
uk	Empirical distribution
lw	Low demand
ir	Unpredictable demand pattern

7.4.2 Results of Lead Time Analysis

As described in Section 4.4, two kinds of methods have been adopted for identifying lead time patterns. The statistical method is used if the historical data of lead time were recorded.

Otherwise, a subjective estimation method will be adopted to obtain the distributions of lead time. Figure 7.11 shows the process of identification of lead time distribution.

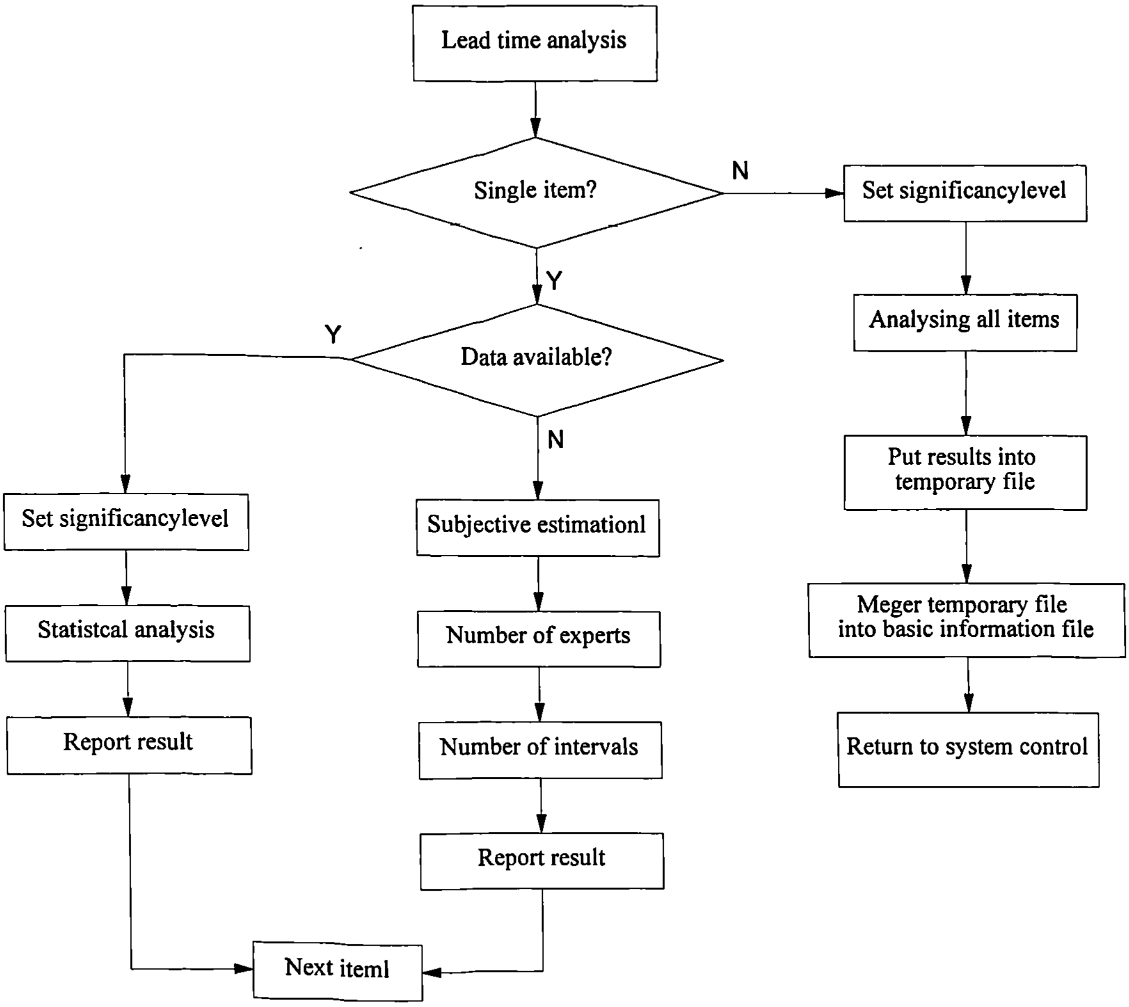
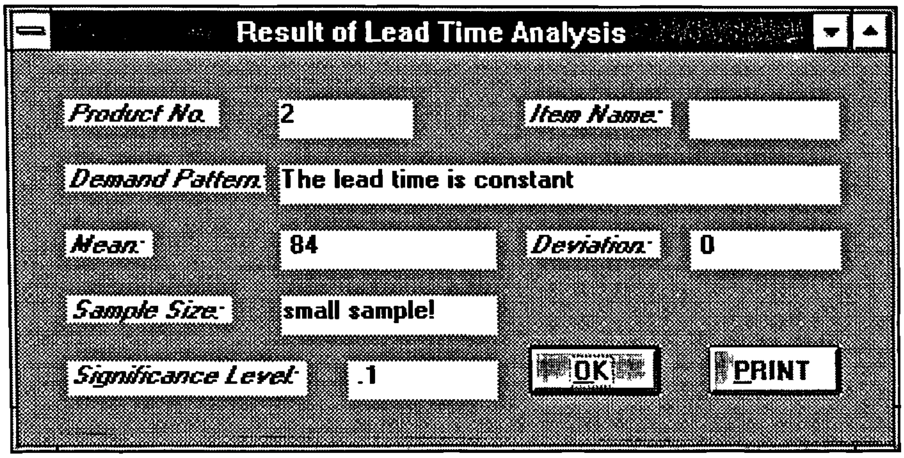


Figure 7.11 The Process of Lead Time Analysis

Using statistical method to identify lead time patterns is similar to the procedure of identifying demand patterns. Fig. 7.12 shows an example of lead time analysis output. However, to obtain the lead time distribution by means of the method of subjective estimation, the user needs to input the number of intervals that the range of lead time has been divided into, and the number of experts who take part in estimating the number of lead times that falls in each interval.



Result of Lead Time Analysis

Product No.	2	Item Name:	
Demand Pattern:	The lead time is constant		
Mean:	84	Deviation:	0
Sample Size:	small sample!		
Significance Level:	.1		

Figure 7.12 The Example Output of Lead Time Analysis

7.4.3 Impact of Significance Levels on the Analysis Outputs

The significance level (α) of a statistical test is the critical value that shows with $(1-\alpha)\times 100\%$ confidence to accept or reject a testing hypotheses. To test the same data series using different values of the significance level (α) for the testing method, the analysis outputs may be different. To take *Part No. 3* as an example (see Fig. 7.10), the output of the system will show that the demand has a Poisson distribution if we change the significance level of constant test from 0.05 to 0.10. Table 7.11 gives the system's outputs based on the testing of 10,000 items. In this table, case 1 and case 2 show change in numbers of items in trend, Poisson, empirical, and unpredictable demand patterns when the significance levels have been changed from 0.05 to 0.1 for these test methods. These results shows how the system's outputs are affected by the significance levels specified to different test methods. Case 3 shows a small reduction of the numbers of items in low demand if the specified critical values for defining low demand pattern has been reduced from 7 to 6. Comparing case 4 and Case 5 we can find the impact on the numbers of items falling into strictly constant (changed from 0 to 844) , quasi constant (from 2146 to 966), and statistically unpredictable (from 428 to 237) categories if we change both the critical values (from 0.1 to 0.5) specified to classify strictly constant and the significance levels (from 0.05 to 0.1) for testing randomness, trend, Poisson, and empirical. Case 5 and case 6 illustrate that only the number of items falling into strictly constant and quasi constant two categories will be affected and the

Table 7.11 The Impact of Different Significance Levels (or critical values) on the Results of Patters Identification

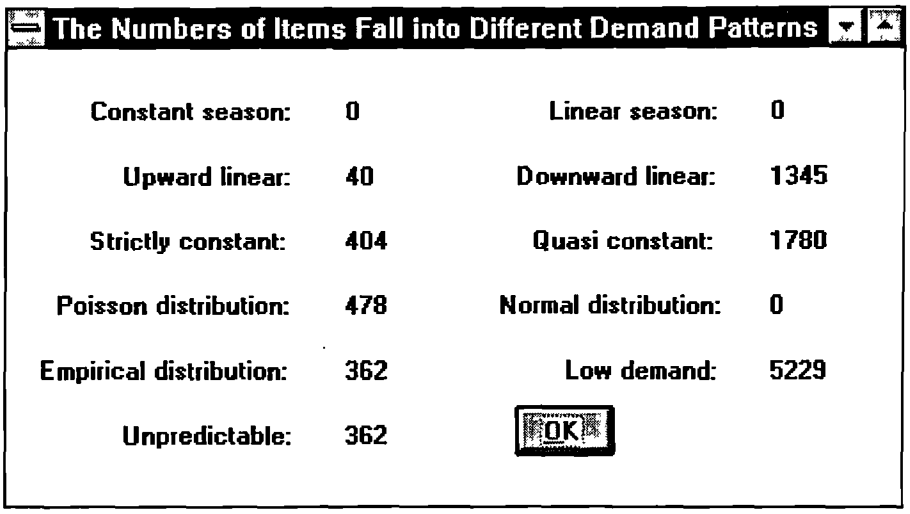
Patterns		Low demand	Seasonal	Strictly cons.	Quasi cons.	Trend	Normal	Poisson	Empirical	Unpredictable
Case 1	Critical value	a < 7	0.05	σ < 1.0	σ < 0.5	0.05	0.05	0.05	0.05	
	No. of items	5229	0	1004	1101	1385	0	487	362	441
Case 2	Critical value	a < 7	0.1	σ < 1.0	σ < 0.5	0.1	0.1	0.1	0.1	
	No. of items	5229	0	808	966	1968	0	300	301	428
Case 3	Critical value	a < 6	0.1	σ < 0.1	σ < 1.5	0.05	0.05	0.05	0.05	
	No. of items	5176	0	0	2350	1392	0	483	362	441
Case 4	Critical value	a < 6	0.05	σ < 0.1	σ < 1.0	0.05	0.05	0.05	0.05	
	No. of items	5176	0	0	2146	1392	0	483	362	237
Case 5	Critical value	a < 6	0.05	σ < 0.5	σ < 1.0	0.1	0.1	0.1	0.1	
	No. of items	5176	0	844	966	1984	0	301	301	428
Case 6	Critical value	a < 6	0.1	σ < 0.1	σ < 1.0	0.1	0.1	0.1	0.1	
	No. of items	5176	0	0	1810	1984	0	301	301	428

a ---- annual demand; σ --- stadard deviation.

number of items falling into other categories remain the same values if we change the critical value for strictly constant test (from 0.5 to 0.1). This means that the number of items falling into strictly constant category will increase and the number of items falling into quasi constant category will decrease by same numbers (1810 = 844 + 966) when the critical value becomes larger, and vice versa.

7.4.4 Time Period to Repeat Patterns Identification

As described in Chapter 3, the system has the ability to update the demand and lead time data files when new transaction records are available. This does not mean that we will repeat the analysis process as soon as any new data are entered into the system. In fact, if we carry out analysis too frequently any recent volatile demand may cause unfounded selection of a new pattern. On the other hand, if we repeat the analysis at very long intervals the old data may mask emerging demand pattern. Therefore, the user has to balance between these effects when selecting the interval of updating the demand patterns.



The Numbers of Items Fall into Different Demand Patterns

Constant season:	0	Linear season:	0
Upward linear:	40	Downward linear:	1345
Strictly constant:	404	Quasi constant:	1780
Poisson distribution:	478	Normal distribution:	0
Empirical distribution:	362	Low demand:	5229
Unpredictable:	362		

Figure 7.13 Report Window of Counting Items in Different Patterns

In this system, two sub-modules have been developed to help the user to decide how often to repeat the analysis. The first sub-module was designed to partition the (demand) data file into different sizes so that the analysis can be carried out on a different sample sizes. The second sub-module has the ability to calculate the numbers of items which fall into different demand types (see Fig. 7.13).

Table 7.12 The Impact of the Sample Size on the Results of Patters Identification

	Patterns	Low demand	Seasonal	Strictly cons.	Quasi cons.	Trend	Normal	Poisson	Empirical	Unpredictable
Sample size	Critical value	a < 7	0.05	$\sigma < 1.0$	$\sigma < 1.5$	0.05	0.05	0.05	0.05	
15 moth data	No. of items	3881	0	3337	767	592	0	724	432	267
18 moths data	No. of items	4391	0	2511	922	690	0	723	485	278
	Change	490	0	826	55	98	0	1	53	11
	Critical value	a < 7	0.1	$\sigma < 1.0$	$\sigma < 0.5$	0.1	0.1	0.1	0.1	
15 moth data	No. of items	3881	0	3244	619	1324	0	534	383	277
18 moths data	Critical value	4391	0	2247	831	1340	0	457	382	351
	Change	510	0	997	212	16	0	74	1	74

a ---- annual demand; σ ---- stadard deviation.

By combining these two sub-modules the user can partition the (demand) data file and compare the analysis results of different sample sizes to help decide when to update the analysis process. Table 7.12 shows analysis results of items with 15 and 18 months data with different significance levels. It indicates that there are no significant changes in the numbers of items that fall into trend, Poisson, Normal, empirical, unpredictable patterns with change the length of history data used. The numbers of items fall into strictly constant, quasi constant and low demand have changed significantly. However, the criteria of these tests which affect the output of analysis tests are arbitrarily specified by the user. Based on Table 7.12 we can conclude that it may be appropriate to carry out analysis every six months on the given demand data. Further testing is needed to confirm such finding.

Clearly, the interval needed to update the analysis is related to the time period over which the demand data were collected. The outputs based on different data set may lead the user to make a different decision. This example is based on a monthly base of demand data.

7.4.5 Data Requirements

There are some requirements on the amount of data needed to perform demand (or lead time) analysis.

Number of Data Points (observations)

From the statistical point of view, the larger the sample size is, the more accurate the estimations will be and the less subject to error the conclusions drawn from the data will be. To obtain estimations of the parameters which describe the properties of demand and lead time data, the number of the data points or observations must exceed a reasonable amount ($n > 20$). For example, the data used to test the system are in a monthly base. Therefore, at least two years demand data are needed to carry out the patterns identification with the exception of the seasonal test.

As discussed in Chapter 4, the series of specific seasonals which is the moving average calculated based on the length of seasonality is used to test if there is seasonal movements in a data series. However, it is about a year shorter than the series being filtered. So that several (at least three) years' worth of data / observations are required.

Completeness of data

To make the system work properly, a complete set of data, which means that all data points have a reasonable values for each item is definitely needed. A incomplete set of data will cause a system error which is difficult to identify. During the system's verification we found that there are *53* inventory items were recorded as having zero unit price which is obviously not correct and will cause halt of the system during estimating the decision parameters.

Consistency of Data

This system is developed on the assumption that all the demand data (or lead time data) are collected over the same time period. If the time unit between two successive observations is different the outputs of the system will be meaningless.

The basic premise of the testing methods employed in this system is that discernible patterns and relationships exists in data and are stable enough to allow extrapolation into the future.

7.4.6 Method to Speed up the Analysis Process

As illustrated in Fig. 3.1, the demand analysis starts with reading an item's demand data from history data file (File 3) and finish with saving the result to the same item's record in basic information file (File 1). When the basic information file is very large, it takes longer time to locate the position of an item's record. For instance, it takes more than three hours to complete the analysis of the demand data file provided by British Airways which contains more than 10,000 items' records saved in basic information file. To speed up the analysis process, the system was designed to save the results of analysis into a temporary file. A special function has been developed to merger the temporary file into the basic information file when the analysis is completed (Fig. 7.14).

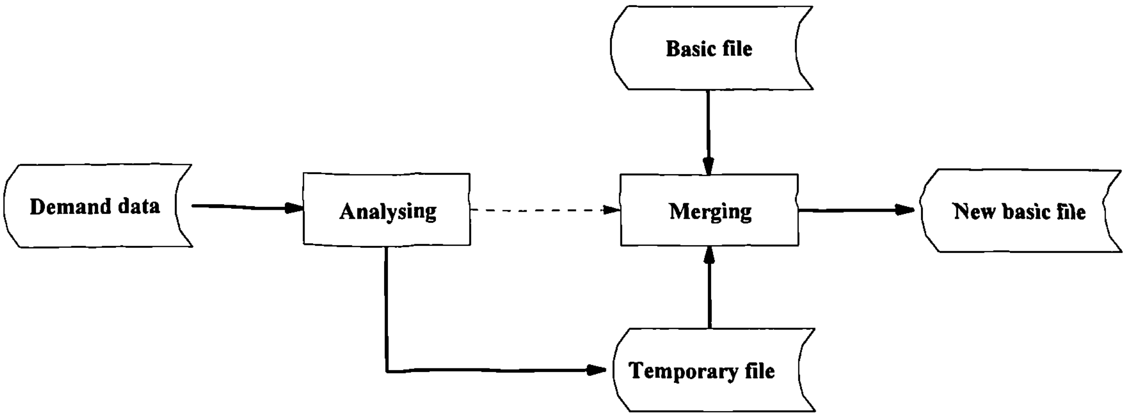


Figure 7.14 Improved Demand Analysis Process

Based on the same data file, the improved demand analysing procedure greatly reduces the time spent on demand analysis to eight and half minutes (see Table 7.13). However, about 43% of time is still used to read from and write to data files because the speed for computers to access to external memory (such as floppy disc, hard disc, and CD-ROM) is relatively slow. The proportion of time spent on different steps of analysis is given in Table 7.13.

Table 7.13 Statistical Results of Time Spent on Demand Analysis (Based on the data supplied by British Airways)

Total time	**Reading** data	**Analysis**					**Save** result
		Low demand	Seasonal	Trend	Probabilisti	Constant	
8'30"	1'20"	37"	2'	49"	20"	14"	3'10"
100%	15.69%	7.25%	23.53%	9.61%	3.92%	2.75%	37.25%

7.4.7 Results of Order Decisions

The order decisions are triggered by different methods. For the items which have a time dependent demand pattern, the order decisions are made at the beginning of the planning horizon. For the items with constant demand and probabilistic demand, the order decisions are made when the current inventory status drops to (or below) their reorder points (Table 7.14). The order quantities are given by the optimal inventory models selected by the rule base based on the above mentioned analysis results and other information such as discount, supply rate, etc.

Table 7.14 Order List

Product Number	Reorder Point	Current Inventory
2	1	1
13	2	1
14	3	3
16	2	0
17	2	2
29	17	5
42	7	4
96	12	6
157	9	6
199	4	1
392	2	0
910	26	16
1550	8	6

The order quantity of an item with constant demand basically depends on annual demand, holding cost, and ordering cost. However, other parameters such as supply rate, planned shortage, and price discount also affect the order decisions. In order to make an order for an item with constant demand pattern, the user may need to input the following information which is prompted by the system:

1. Discount break points(quantity);
2. Discount prices;
3. Shortage cost;
4. Supply rate.

Table 7.15 The Order Decisions of an Item with Constant Demand under Different Situations

Case	Supply rate		Shortage		All uint discount				Icrement discount				Order
No	Infinite	Finite	Cost(£)	Not allowed	Point1	Price1 (£)	Point2	Price2(£)	Point1	Price1(£)	Point2	Price2(£)	quantity
1	yes	-	-	yes	-	-	-	-	-	-	-	-	80
2	-	1916	-	yes	-	-	-	-	-	-	-	-	113
3	-	2897	-	yes	-	-	-	-	-	-	-	-	98
4	yes	-	37.5	-	-	-	-	-	-	-	-	-	87
5	yes	-	75.5	-	-	-	-	-	-	-	-	-	84
6	-	1916	37.5	-	-	-	-	-	-	-	-	-	694
7	-	2874	75.5	-	-	-	-	-	-	-	-	-	600
8	yes	-	-	yes	120	35.5	160	30.5	-	-	-	-	160
9	yes	-	35.5	-	120	35.5	160	30.5	-	-	-	-	120
10	yes	-	75.5	-	120	35.5	160	30.5	-	-	-	-	120
11	-	1916	-	yes	120	35.5	160	30.5	-	-	-	-	160
12	-	2874	-	yes	120	35.5	160	30.5	-	-	-	-	160
13	-	1916	35.5	-	120	35.5	160	30.5	-	-	-	-	767
14	-	2874	75.5	-	120	35.5	160	30.5	-	-	-	-	663
15	yes	-	-	yes	-	-	-	-	100	30.5	120	25.5	80
16	yes	-	35.5	-	-	-	-	-	100	30.5	120	25.5	87
17	yes	-	75.5	-	-	-	-	-	100	30.5	120	25.5	84
18	-	1916	-	yes	-	-	-	-	100	30.5	120	25.5	113
19	-	2874	-	yes	-	-	-	-	100	30.5	120	25.5	98
20		1916	35.5	-	-	-	-	-	100	30.5	120	25.5	667
21	-	2874	75.5	-	-	-	-	-	100	30.5	120	25.5	659

Table 15 shows the order decisions of an item which has the following basic information:

Product-No.	Demand-pattern	Unit-price	Holding-cost	Ordering-cost	Annual-demand
0136	Constant	£37.27	£7.45	£25	958

The order quantity of this item is 80 units if the supply rate is infinite, planned shortage is not allowed, and there is no discount (see Case 1 in Table 15). Cases 2 and 3 indicate that if the supply rate is finite the order quantities will be increased, but the order quantities will decrease when the supply rate is increased (see Equation 5-11). Cases 4 and 5 show that if planned shortage is allowed the order quantities will be increased. However, the order quantities will decrease as the shortage cost increases (see Equation 5-19). In cases 6 and 7, the order quantities are very large because both the finite supply rate and the planned shortage will increase the order quantity (see Equation 5-30). Cases 8 to 21 show the order quantities when all unit discount and increment discount are taken into account.

The inventory status of an item with low demand is dynamic. It can be changed by different events (see Table 7.16). The second row of Table 7.16 shows the current inventory status which will trigger an event (see row 3) to make a new order because its current inventory has dropped below its reorder-point. The order quantity (2 units) will be added to on-order inventory (see column 4 of row 3). The second event is to demand 1 unit of the item. This will decrease the on-hand inventory by 1 unit (see row 4). The event of receiving an order (see row 5) will decrease the on-order inventory and increase on-hand inventory by the same amount (2 units). Row 6 shows a demand event which reduces the on-hand inventory by 1 unit of the item. At the same time, this demand event also triggered an order event which will add 2 units to on-order inventory.

Table 7.17 shows the dynamic inventory status of an item with probabilistic demand. The changing process of the inventory status of an item which has probabilistic demand pattern is similar to the dynamic inventory status of an item with low demand because same inventory policy is adopted to treat these two kinds of inventory problems. The order event which happens when the current inventory drops below the reorder-point will increase the on-order inventory (see rows 3 and 7). The demanding event triggered by customers will reduce the on-hand inventory which may invoke ordering event (see rows 4 and 6). The receiving of an order event will reduce the on-order inventory and increase on-hand inventory by the same amount at the same time (see row 5).

Table 7.16 The Dynamic Inventory Status of an Item with Low Demand

Event	Reorder-point	On-hand	On-order	Backorder	Maximum
Current status	2	1	0	0	3
Order *2* units	2	1	2	0	3
Demand *1* unit	2	0	2	0	3
Receive *2* units	2	2	0	0	3
Demand *1* unit	2	1	0	0	3
Order *2* units	2	1	2	0	3

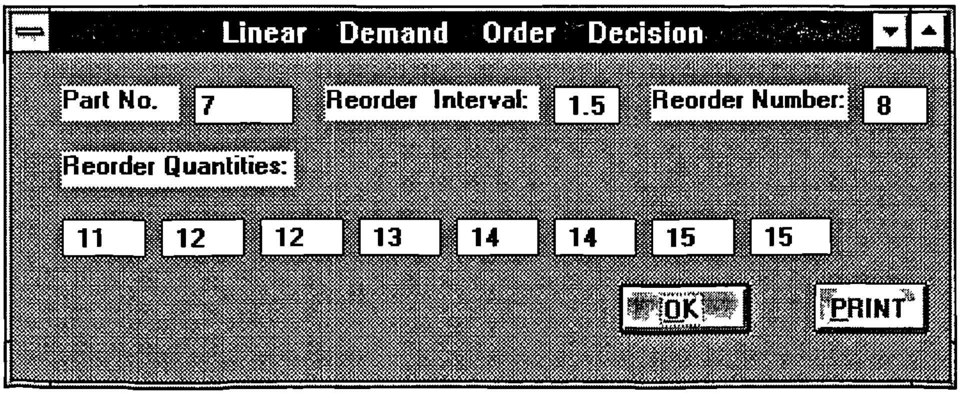
Table 7.17 The Dynamic Inventory Status of an Item with Probabilistic Demand

Event	Reorder-point	On-hand	On-order	Backorder	Maximum
Current status	24	14	0	0	69
Order 55 units	24	14	55	0	69
Demand 20 units	24	0	55	6	69
Receive 55 units	24	49	0	0	69
Demand 30 units	24	19	0	0	69
Order 50 units	24	69	19	50	69

Table 7.18 shows some of the order decisions of items with trend demand pattern. The reorder interval (in month) is constant which is very easy for inventory managers to control. The number of reorders indicates how many orders will be taken place during the panning horizon. Fig. 7.15 illustrates the system's output of order decision for an item with increasing demand. In this case it is assumed that the lead time is constant.

Table 7.18 Summary of Order Decisions of Items with Trend Demand

Part	Order	No. of	Order Series and Quantity							
No.	Interval	Order	1	2	3	4	5	6	7	8
7	1.5	8	11	12	12	13	14	14	15	15
9	2.7	5	132	147	161	176	61			
45	6.2	2	8	9						
58	1.9	6	4	4	4	4	4	6		
62	3.7	4	23	27	30	8				
64	7.9	2	9	6						
71	20	1	11							
103	8.9	2	7	3						
125	12	1	20							
133	3	4	29	32	35	36				


Linear Demand Order Decision

Part No. 7

Reorder Interval: 1.5

Reorder Number: 8

Reorder Quantities:

11 12 12 13 14 14 15 15

Figure 7.15 Ordering Decision of an Item with Increasing Demand

The proportion of times different inventory models were used by the IDSS is shown in Table 7.19. No item with seasonal demand appeared in the data sets given by the BA. A choice of models to use occurs when an item's demand is quasi constant. In which case the system prompts suggested options for the user to choose. This choice will presumably be influenced by how important an item is. If the item is very important and needs to be effectively controlled, the probabilistic model should be applied. Otherwise, the constant model will be applied to simplify inventory management tasks. Table 7.20 also indicates that when only 12 months of data are used, almost a third of items have statistically unpredictable demand patterns as compared with 3.6% when 24 months of data are used. This demonstrates the importance of an appropriate data set for model support.

Table 7.19 The Proportion of the Inventory Models Being Used (based on 24 months demand data file supplied by British Airways)

Pattern	Low demand	Seasonal	Trend	Probabilistic	Constant	Unpredictable
100%	52%	0.0%	14%	8.4%	22%	(3.6%)

Table 7.20 The Proportion of the Inventory Models Being Used (based on 12 months demand data supplied by British Airways)

Pattern	Low demand	Seasonal	Trend	Probabilistic	Constant	Unpredictable
100%	24.3%	0.0%	0.002%	12.1%	32.2%	31.4%

Table 7.21 indicates that substantial cost savings might be made by reducing inventory levels. Although it is not anticipated that inventory managers would be likely to adopt the new system immediately, the results show the direction of expected improvements. In practice, the new system will be run in parallel with existing systems for some time to prove its efficiency by comparing the performances of the two systems. The accuracy and effectiveness of the new system will be improved as more data is collected and enter into the system.

Table 7.21 Comparing Results

Part number	Unit price	Average Inventory (recommended)	Average Inventory (current system)	Savings of inventory investment (%)
2	11.18	1	2	50
6	25.53	1	7	85.7
10	15.78	1	1	0
26	162.98	3	4	25
101	3.2	7	5	-40
109	0.51	23	17	-35.2
140	10	3	10	70
161	285	1	5	80
238	7.39	4	22	81.8
261	35.00	3	2	-50
299	690.77	1	4	75
340	4.26	5	8	37.5
396	14.59	6	3	-66.6
410	1.00	10	21	52.4
446	3.00	7	9	22.2
493	5.11	18	20	10
505	3.63	20	18	-11.1
541	9.84	9	32	71.87
622	1.00	12	16	25
641	5.00	5	5	0
677	2.00	8	15	46.7
712	3.00	9	18	50
725	6.97	7	4	-75
755	1.98	12	26	53.8

Chapter 8

CONCLUSIONS AND FURTHER WORK

The purpose of the work described in this thesis is primarily to demonstrate the method of applying and the usefulness of using the KBS approach in the optimisation of large scale independent demand inventories. This chapter provides an overview of the complete research project and its conclusions and outlines further directions to extend and advance the current research.

8.1 Conclusions

Generally practitioners in OR have used classical mathematical approaches for modelling and decision making. These provide firm theoretical foundation and often lead to optimal solutions. However, real world problems are often too vague, too ill-structured, or too complex for formal mathematical modelling [83]. The complex nature of the problems encountered today in business and industry, and the competitive environment in which they exist, requires the processing of increasing volumes of data in increasingly complex, innovative and efficient ways [168]. Recent advances in MS/OR and ES techniques provide tools enabling this need to be addressed. Consequently there has been a marked advance in the use of knowledge-based systems to support management decision making in different areas of business and industry (see Section 2.2.4). This thesis presents an intelligent inventory management system which is able to obtain history data from existing inventory database, identify demand patterns based on these data, and help inventory managers in making decisions using mathematical inventory models selected from the model base by the rules contained in the rule base in according to the results of data analysis and the characteristics of the practical situation in hand. It is hoped that the system would promote the applications of published inventory models, and help inventory managers to achieve efficient and effective performance of inventory management.

The development of a prototype knowledge-based intelligent system embodying these concepts has been described and some results from its application to real inventory history data have been discussed. The approach was found to be very useful and to have a great deal of further potential in practical application, especially in the classification and choice of models. This can be valuable achievement because inventory problems are common to all business/industry organisations including: retailers, wholesalers, distributors, and manufacturers.

The following are the achievements and conclusion of this research:

1. An inventory KBS has been developed. The system consists of six major components including pattern identifier, data manager, model base, rule base, user interface, and on-line help system. It has the unique feature of being able to identify the demand data pattern and hence automatically select a suitable inventory model.

Differing from other inventory KBSs published in the literature, the development of this system is based on practical situations and co-operation with two leading manufacturing and service organisations; Optical Fibres and British Airways. The system's verification was carried out using the real data provided by the companies and the test results (see Table 7.15) show that there is a scope for great potential savings in inventory investment as a result of applying this system.

2. The developed pattern identifier is integrated into the system to serve as a front-end interface, and combines logical and statistical methods to identify data pattern of independent demand inventory items from the history data without the need to interrogate the user. No previous attempts to develop such a subsystem were found in

the literature of applying KBSs in inventory management. With this system, a user can analyse demand patterns without having knowledge of the theory of statistics. But a user who has some statistical knowledge can run it more effectively by adjusting the test criteria instead of accepting the system default parameters. Although this pattern identifier is designed to identify demand patterns, it is a relatively independent module and can be used in identifying patterns of other parameters.

3. To facilitate the function of the pattern identifier, a data manager has been specially developed to process the demand history data as well as other basic inventory information, used for calculating the order quantity, because the system's performance is mainly dependent on the integrity and quality of data. The data manager can also read data stored in other applications. This is a very important factor in integrating the system into existing computer-based inventory management systems, which makes the system more acceptable by inventory managers, and promote its application in the real world.

4. A model base, mainly concerned with single item optimisation, has been developed. The model base consists of inventory models dealing with constant demand, stochastic demand, slow-moving items, demand with linear trend, and demand with cyclical fluctuation. To establish the system's model base, the existing models dealing with similar inventory problems have been compared based on its applicability, clarity, and being suitable to be computerised. Some new models handling situations of constant demand with discount, shortage, and restricted supply rate and seasonal fluctuation have been developed. Modification was also made to Mitra's et al [145] model for demand with linear trend to cope with the situation where the number of reorders is not an integer over the planning horizon.

5. The rule base of this system provides the required expertise for choosing a suitable inventory model to apply to a real world situation based on the results of

statistical analysis (pattern identification) and the user's response to the system. The rule base can be easily amended or expanded without the need to redesign the whole software, as it is the case with constructural traditional software.

6. To build up the communication channel between the user and the system, a graphical user interface has been developed by means of the UGI facilities provided by Visual Basic. The user interface includes several important capabilities such as the ability to capture data from the user, data presentation through a range of output devices, windowing, graphical presentation and providing data exchange channels with other applications.

7. The help system provides the user with on-line information about how to operate the inventory IDSS. This facility of the system combined with its user friendly interface makes the system easy to use.

8.2 Further Work

The work presented in this thesis has some limitations which should be investigated and overcome.

- First, the system was developed, tested, and modified based on the data and management practices of only two companies. The model base and the rule base needs to be extended in order to fit complicated inventory problems.

- Second, the system lacks the ability of adding new rules and models to the knowledge base without changing the original program because Visual Basic 3.0 which was used to develop the system does not have the facility to add this feature (VB 4.0 which is available now, has the facility to manage rule base).

- Third, as mentioned in Chapter 1, the research is limited to the single-item, single-location, independent demand inventory problems.

- Finally, the system has not been installed to manage real world inventory. Although the system has been examined and experimented using real data, a practical installation will be useful in consolidating, improving, and validating the system to ensure the function and performance characteristics are accepted. It is virtually impossible for a software developer to foresee how the user will really use a program. Instructions for use may be misinterpreted, strange combinations of data may be regularly used, and output that seems clear to the developer may be unintelligible to a user in the field. Therefore, from the practical point of view, an acceptance test conducted by the end user rather than the system developer over a period of months will be very useful.

From the above remarks one can draw the conclusion that there are two main directions for the future work. The first direction is concerned with the techniques of implementing the KBS approach. The second direction is concerned with the consolidation of the knowledge base of the system. The major areas for further work are:

1. Business and management professionals are already acquainted with the tabular format used in spreadsheet application. It would be very useful to build a developing tool to automatically generate decision rules with spreadsheet format (see Table 8.1). To add new rules to the knowledge base of the system, users only need to fill in the rule generation table instead of changing the original code. Rules generated by the tool are:

Rule 1: If *A* and *B* Then *C*; Rule 2: If *A* and *D* Then *F* and *B;*
Rule 3: If *B* Then *G* and *H;* Rule 4: If *G* and *D* Then *H.*

Table 8.1 Format for Generating Rules

Rule Number	Condition i	Condition ii	Action i	Action ii
1	A	B	C	
2	D	E	F	B
3	B		G	H
4	G	D	H	

2. The system was designed for a single user. If it is extended for multiple users, problems of concurrent control may occur [169]. therefore, the system must be capable of ensuring that the database (which contains the history data for generating inference facts and the data for monitoring inventory status and calculating order quantity) is updated correctly when two or more users are involved in making updates to the database concurrently.

To illustrate the problem involved in concurrent update, let us assume user A and user B are working on the same item which has inventory of 150 units. Now user A is accessing the database to receive 100 units from a supplier which will increase the item's inventory level. At the same time, user B is accessing the database to issue 100 units to satisfy a customer demand. The item's inventory level should still be 150 units because the amount of the increase exactly matches the amount of the decrease. However, when user A and user B read the data from the database into their working station respectively, they both have the same data for the item's inventory. At this point, user A adds 100 units to the item's inventory in user A's work station, and user B subtract 100 units from the item's inventory in user B's work station. At this moment if user A writes to the database, the item's inventory becomes 250 units. Then user B writes to data base to change the item's inventory into 50 units. Apparently, the system has saved incorrect data into the database. The reverse order of updates will also give an incorrect result.

One way to prevent this situation from occurring is to prohibit the opportunity for a concurrent update, i.e. the user would be able to read data from the database, but if they enter some data to update the database, the data base would not really be updated at all. Instead, the data would be placed in a separate temporary file. Periodically, a single update program would read the data from the temporary file and acutely perform the appropriate updates to the database. Since this program is the only program allowed to modify the database, it will overcome the concurrent problem.

3. It is well known that inventory modelling is the most developed area in MS/OR. There is considerable scope for further work on the modelling and statistical analysis incorporated in this system although the practical value of this depends on the availability of data of appropriate accuracy and integrity. First, the model base should be enriched to include the multiple items and multiple location inventory problems. Second, heuristics or models for inventory items with statistically unpredictable demand patterns should be investigated as there are plenty of theoretical and practical work that may have to be conducted in this area. As commented by New et al [170], OR should focus more on increasing the sophistication of its models of real world activities than on building more complex models of hypothetical worlds. The existing inventory models for solving similar inventory problems should be compared and improved from the practical point of view to make them workable for the inventory managers.

4. Neural networks in the wide sense are among the most rapidly developing scientific fields today. The rapid rise in popularity and productivity of this field has been mainly due to the success of these networks in performing relatively low-level functions. These functions include perception, segmentation, and classification of patterns [171]. The main advantages of this technology are its ability to adapt to the characteristics of example data, and classify it correctly among noise, distortions and non-linearity and the ability to handle data from multiple sources and decision systems

[172]. Neural networks are very easy to integrate with existing domain knowledge. Therefore it can be used as a front-end interface to the system for identifying inventory demand patterns. But one of the drawbacks is that a number of samples with results known beforehand will be used in training algorithms.

5. It has been acknowledged in this study that most industrial organisations do not maintain the integrity of history data. For example, the ordering costs and holding costs were not recorded at all, and that is a serious limitation for the application of KBSs. It is vital to convince those people within the organisations responsible for data gathering that recorded data will actually be used to improve the inventory management performance and motivate them to improve the quality and scope of history data during their day to day work.

6. There are two key ways to acquire knowledge for building the knowledge base of an intelligent decision systems. One way is to abstract accumulated knowledge from papers and textbooks, the other way is to draw decision rules from the practitioner's experience. This study focuses on using standard knowledge to carry out standard inventory problems. Therefore, the decision rules and model base of this system only consist of the knowledge gathered from academic literature. Further work is needed to combine these methods and include the inventory managers' real expertise to enrich the system's knowledge base. In the current system, the expert manager interacts with the system more than inputs to it other than through parameter setting.

Appendix 1

MANUAL FOR OPERATING THE SYSTEM

1. Commands

1.1 Commands for the System

File Menu

Create data file .... Create a new data File.

Rename data file .... Renames an existing data file.

Delete data file .... Deletes an existing data file.

Recover data file .... Recover a deleted data file.

Merge data file .... Merge the data files which contain the analysis results.

Input data

Basic information .... Input the basic information of an inventory item.

Usage data .... Input the historical demand data.

Lead time .... Input lead time data.

Exchange data .... Load data form other application.

Update data

Basic information .... Modify the basic information of an inventory item.

Usage data.................... Modify the historical demand data.

Lead time .... Modify the lead time data.

Analysis result .... Update analysis results.

Receive goods .... Increase the stock level with received order.

Issue goods .... Reduce the stock level with issued quantity.

Display item's information

Basic information .... Display the basic information of an item.

Demand data .... Display the demand data.

Lead time .... Display the lead time data.

Print item's information

Basic information .... Print the basic information of items.

Demand data .... Print demand data.

Lead time data .... Print lead time data.

Delete an item .... Erase all information of an item from relative data files.

System Initiate ...., Configure the system.

Previous Menu .... Switches to previous window.

Exit ....Quits the system.

Analysis Menu

Demand pattern

Single item .... Identify the demand pattern of an item.

All items..................... Identify the demand patterns of all items.

Lead time pattern

Single item

Statistic methods.........Using statistic methods to identify lead time pattern.

Subjective methods.....Using statistic methods to identify lead time pattern.

All items .... Identify the lead time patterns of all items.

Set reorder point .... Calculate the reorder point.

Set maximum inventory ....Calculate the maximum inventory level.

Decision Menu

Order Decision .... Make a new order for an item.

Order list .... Display the items that need replenishing.

Utilities Menu

Calculator .... Loads Calculator.

Calendar .... Loads Calendar.

Cardfile .... Loads Cardfile.

Excel .... Loads Microsoft Excel.

Word .... Loads Microsoft Word.

Count items .... Count the number of different demand patterns.

Data file partition .... Divide data file into different sizes.

Help Menu

Index .... Displays contents of the help system.

How to use help .... Displays information for using the help system.

About the system ....Display the brief description of the system.

1.2 Commands for the Help

File Menu

Print topics .... Print help information which you selected.

Exit .... Quit help and switches to system.

Edit Menu

Copy .... Copy selected help information to the Clipboard.

Annotate .... Add new information to selected help topic.

Help Menu

Display how to use help information.

2. How to Carry Out a Procedure

Create Data File

1. Click the "Create Data File" from File menu.
2. Enter a file name.
3. Click "OK" to create a new data file.

Then the file name which you created will be displayed on screen.

Click "CANCEL" button to quit.

Rename Data File

1. Click the "Rename Data File" from File menu.
2. Click "Driver List" to select driver.
3. Click "Directory List" to select directory.
4. Click the file which you want to rename from "File List".
5. Enter a new name .
6. Click "RENAME" button to rename the file.

Then both the new and old file name will be displayed on screen.

Click "CANCEL" button to quit.

Delete Data File

1. Click "Erase Data File" from File menu.
2. Click "Driver List" to select driver.
3. Click "Directory List" to select directory.
4. Click the file which you want to rename from File List.
5. Click "DELETE" button to delete the file.

Then confirming information will be displayed on screen.

Click "CANCEL" button to quit.

Recover Data File

1. Click the "Recover a Deleted File" from File menu.
2. Click "Directory List" to select directory.
3. Click the file which you want to recover from "File List".
4. Enter a new name .
5. Click "RECOVER" button to recover the file.

Then both the new and old file name will be displayed on screen.

Click "CANCEL" button to quit.

Merge Data File

1. Click "Merge Data Files" from File menu.

Then the data files contain the analysis results will be merged into basic information file.

Input Basic Information

1. Click "Basic information" from Input sub menu.
2. Input available number of data.
3. Input data.
4. Click ENTER button to store the data in data file.

Click "CANCEL" button to quit.

Input Demand Data

1. Click "Usage data" from Input sub menu.
2. Input available number of data.
3. Input data.
4. Click ENTER button to store the data in data file.

Click "CANCEL" button to quit.

Input Lead Time Data

1. Click "Lead time" from Input sub menu.
2. Input available number of data.
3. Input data.
4. Click ENTER button to store the data in data file.

Click "CANCEL" button to quit.

Load Data from Other Application

1. Click "Exchange Data" from Input sub menu.
2. Input data file name
3. Input the number of data requested.
4. Click "OK" button to load the data into a Visual Basic data file.

Click "CANCEL" button to quit.

Modify Basic Information

1. Click "Basic information" from Update sub menu.
2. Modify data.
3. Click "MODIFY" button.

Click "CANCEL" button to quit.

Update Demand Data

1. Click "Usage data" from Update sub menu.
2. Input the number of new available data.
3. Input data.
4. Click "OK" button.

Click "CANCEL" button to quit.

Update Lead Time Data

1. Click "Lead time" from Update sub menu.
2. Input the number of new available data.
3. Input data.
4. Click "OK" button.

Click "CANCEL" button to quit.

Input Received Order

1. Click "Receive goods" from Update sub menu.
2. Input the receive amount.
3. Click "OK" button to increase stock level.

Click "CANCEL" button to quit.

Input Issued Quantity

1. Click "Issue goods" from Update sub menu.
2. Input the issued amount.
3. Click "OK" button to reduce stock level.

Click "CANCEL" button to quit.

Display Item's Basic Information

1. Click "Basic information" from Display sub menu.
2. Choose "Display Option" to show items one by one or just single item.
3. Click "YES" to display next item.

Click "NO" button to quit.

Display Demand Data

1. Click "Demand data" from Display sub menu.
2. Choose "Display Option" to show items one by one or just single item.
3. Click "YES" to display next item.

Click "NO" button to quit.

Display Lead Time Data

1. Click "Lead time" from Display sub menu.
2. Choose "Display Option" to show items one by one or just single item.
3. Click "YES" to display next item.

Click "NO" button to quit.

Print Item's Basic Information

1. Click "Basic information" from Print sub menu.
2. Choose "Y" to print all items.
3. Choose "N" to print single item.

Click "OK" without entering "Y" or "N" to quit

Print Demand Data

1. Click "Demand" from Print sub menu.
2. Choose "Y" to print all items.
3. Choose "N" to print single item.

Click "OK" without entering "Y" or "N" to quit

Print Lead Time Data

1. Click "Lead Time" from Print sub menu.
2. Choose "Y" to print all items.
3. Choose "N" to print single item.

Click "OK" without entering "Y" or "N" to quit

Delete an Item

1. Click "Delete Item" from File menu.
2. Input the product number which need to be deleted.
3. Click "DELETE" button to delete the item which is display on screen.

Click "CANCEL" button to quit.

Initiate System

1. Click "System Initiation" from File menu.

3. Input parameter's new values or accept default values.

Click "OK" to configure the system.

Identify Demand Pattern of Single Item

1. Click "Demand Pattern" from Analysis menu.
2. Click "Single Item" of the "Demand Pattern" sub menu.
3. Enter the product number.
4. Click "OK" to carry out the analysis.

Click "CANCEL" button to quit.

Identify Demand Patterns of All Item

1. Click "Demand Pattern" from Analysis menu.
2. Click "All Items" of the "Demand Pattern" sub menu.

Click "OK" button to return when the job is done.

Identify Lead Time Pattern of Single Item

1. Click "Lead time" from Analysis menu.
2. Click "Single Item" of the "Lead Time Pattern" sub menu.

2. Enter the product number.

3. Click "OK" to carry out the analysis.

Click "CANCEL" button to quit.

Identify Lead Time Patterns of All Items

1. Click "Lead Time Pattern" from Analysis menu.

2. Click "All Items" of the "Lead Time Pattern" sub menu.

Click "OK" button to return when the job is done.

Set Reorder Point

1. Click "Set reorder point" from Analysis menu.

The system will carry out process automatically.

Set Maximum Inventory

1. Click "Set maximum inventory" from Analysis menu.

The system will carry out the process automatically.

Make a New Order

1. Click "Order decision" from Decision menu.

2. Input item number.

3. click "OK" to make order decision.

Click "CANCEL" button to quit.

List All Items That Need Replenishment.

1. Click "Order list" from Decision menu.

The system will carry out the process automatically.

Start Calculator

Click "Calculator" from Utility menu.

To return exit the Calculator.

Start Excel

Click "Excel" from Utility menu.

To return exit the Excel.

Start Calendar

Click "Calendar" from Utility menu.

To return exit Calendar.

Start Cardfile

Click "Cardfile" from Utility menu.

To return exit Cardfile.

Start Microsoft Word

Click "Word" from Utility menu.

To return exit Microsoft Word.

Count Items with the Same Patterns

Click "Count Items" from Utility menu.

The system will count the number of items having the same demand patterns.

Data File Partition

Click "Data File Partition" from Utility menu.

The system will divide the data file into different sizes.

Access Help Information

Press F1 or click "Contents" from Help menu.

Print Help Topic

Click "Print Topics" from help's File menu..

Copy Help Topic to Clipboard

Click "Copy" From help's Edit menu.

Annotate Help Information

1. Click "Annotate" from help's Edit menu.
2. Enter attaching information.

3. Click "ENTER" button to add the information to the help topic.

Click "CANCEL" button to quit.

List Help Topic Which You Have Accessed

Click "History" button.

Switch to Previous Help Topic.

Click "Back" Button.

Display Help Contents

Click "Contents"

Exit Help.

Click "Exit" form help's File menu.

System Initiation

1. Click "System Initiation" from File menu.

Switch to Previous Menu

1. Click "Previous Menu" from File menu.
2. Input the values of the parameters for configuring the system.

Exit System

Click "Exit" form File menu.

3. Shortcut Keys

To run the system with keyboard, the following shortcut keys will be used.

3.1 To Open a Top Level Menu.

Press:	To:
Alt + F	Access File menu.
Alt + D	Access Analysis menu.
Alt + E	Access Decision menu.

Alt + U	Access Utilities menu.
Alt + H	Access Help menu.

3.2 To Access Commands in a Menu.

3.2.1 When the File menu is opened by press Alt + F, press the following keys will access the correspondent commands.

Press:	To:
Shift + C	Create a new Data File.
Shift + R	Rename an existing data file.
Shift + E	Delete an existing data file.
Shift + V	Recover a deleted file.
Shift + M	Merge data files
Shift + U	Update data.
Shift + S	Display an item's properties.
Shift + T	Print an item's information.
Shift + D	Delete an item's properties.
Shift + Y	Initiate the system.
Shift + P	Go back to previous window.
Shift + X	Quit the system.

3.2.2 When the Analysis menu is opened by press Alt + A, press the following keys will access the correspondent commands.

Press:	To:
Shift + P	Identify demand pattern.
Shift + L	Identify lead time distribution.
Shift + S	Set reorder point.
Shift + E	Set maximum inventory level.

3.2.3 When the Decision menu is opened by press Alt + E, press the following keys will access the correspondent commands.

Press:	**To:**
Shift + D	Order decision.
Shift + L	Order List.

3.2.4 When the Utility menu is opened by press Alt + U, press the following keys will access the correspondent command

Press:	**To:**
Shift + C	Load Calculator.
Shift + A	Load Calendar.
Shift + R	Load Cardfile.
Shift + E	Load Microsoft Excel.
Shift + W	Load Microsoft Word.
Shift + O	Count the number of items having the same pattern.
Shift + D	Divide a data into different sizes.

3.2.5 When the Help menu is opened by press Alt + H, press the following keys will access the correspondent commands.

Press:	**To:**
Shift + I or F1	Display contents for help.
Shift + H	Display information for using help.

3.3 Shortcut Keys for Using Help

3.3.1 Open a help's top level menu

When the help window is opened, press the following keys will access the help's top level menus.

Press:	**To:**
Alt + F	Access help's File menu.
Alt + E	Access help's Edit menu.
Alt + H	Display the information about how to use the help.

3.3.2 Access the Commands in a Menu

3.3.2.1 When the help's File menu is opened by press Alt + F, press the following keys will access the correspondent commands.

Press:	**To:**
Shift + P	Print help information which you selected.
Shift + X	Quit help and switch to system.

3.3.2.2 When the help's Edit is opened by press Alt + E, press the following keys will access the correspondent commands.

Press:	**To:**
Shift + C	Copy the selected help information to the Clipboard.
Shift + A	Add new information to a selected help topic.

If there is symbol "▸" after a command, click this command will open a sub menu. When a sub menu is opened, a command will be performed by pressing the underline letter of this command.

Appendix 2

THE DERIVATION OF SOME MODELS FOR CONSTANT DEMAND

As discussed in Chapter 5, the models for constant demand (including Q6, Q9, Q10, and Q12) which contain more than one factor are not available in the literature and have to be developed to minimise the total annual inventory costs for these situations. The details of the development of these models are given in this appendix and the same notation used in Section 5.2.1 is adopted for developing these models, i.e.

B = Amount of shortage;

$B^{(i)}$= Amount of shortage at break point *i;*

C_h = Holding cost;

C_p= Purchasing cost;

C_s = Shortage cost;

$C_h^{(i)}$= Holding cost of different discount break points;

$C(q_i)$= Expenditure of purchasing an item for q_i units ($C(q_0) = 0$);

D= Annual demand;

$i = 0, 1, 2, ..., m;$

I = The cost of having one pound of item stored for a unit time;

m = Number of discount break points;

q_i = Quantity of different discount break points ($q_0 = 0$);

$Q^{(i)}$= Order quantity according to break point *i*;

R_s= Supply rate;

U_p= Unit price;

$U_p^{(i)}$= Prices of different discount break points;

$U_p^{(0)} = U_p$.

1. The Derivation of the Model for Finite Supply Rate and Shortage Allowed (Q10)

In this case, the cost components are computed as follows:

$$\text{Purchasing cost} = DU_p + \frac{D}{Q}C_p$$

$$\text{Holding cost} = \frac{C_h}{2}(Q(1-\frac{D}{R_s}) - B)(1-\frac{B}{Q}-\frac{BD}{Q(R_s-D)})$$

$$\text{Shortage cost} = \frac{C_sB^2}{2Q}(\frac{R_s}{R_s-D})$$

To combine these cost components, we get the total annual cost, that is

$$TC(Q,B) = DU_p + \frac{D}{Q}C_p + \frac{C_sB^2}{2Q}(\frac{R_s}{R_s-D}) + \frac{C_h}{2}(Q(1-\frac{D}{R_s}) - B)(1-\frac{B}{Q}-\frac{BD}{Q(R_s-D)})$$

Taking the partial derivative of the equation with respect to Q and B, we have

$$\frac{\partial\, TC(Q,B)}{\partial\, Q} = -\frac{D}{Q^2}C_p - \frac{B^2C_sR_s}{2Q^2(R_s-D)} + \frac{C_h}{2}(\frac{R_s-D}{R_s} - \frac{B^2}{Q^2} - \frac{B^2D}{Q^2(R_s-D)})$$

and

$$\frac{\partial\, TC(Q,B)}{\partial\, B} = \frac{BC_sR_s}{Q(R_s-D)} + \frac{C_h}{2}(-2+\frac{2B}{Q}+\frac{2BD}{Q(R_s-D)}).$$

Setting each equation to zero and solving for Q, we have the optimal order quantity

$$Q10 = Q = \sqrt{\frac{2DC_p}{C_h}\cdot\frac{R_s}{R_s-D}\cdot\frac{C_s+C_h}{C_s}}$$

and the quantity of shortage

$$B = \frac{C_h(R_s - D)}{R_s(C_s + C_h)} Q.$$

2. The Derivation of the Model for Increment Discount

Let the quantities $q_{1,}\ q_{2,\ldots\ldots,}\ q_{m:}$ be the discount break points. If quantity $Q^{(i)}$ is purchased ($q_i < Q^{(i)} < q_{i+1}$ that means $Q^{(i)}$ never happens at the discount break points [110], The total cost of $Q^{(i)}$ units is

$$C(Q^{(i)}) = C(q_i) + U_p^{(i)}(Q^{(i)} - q_i)$$

where

$$C(q_0) = 0; \qquad and \qquad q_0 = 0.$$

The average cost per unit is

$$\frac{C(Q^{(i)})}{Q^{(i)}}) = \frac{C(q_i)}{Q^{(i)}} + U_p^{(i)} - \frac{q_i}{Q^{(i)}} U_p^{(i)})$$

In these cases, the procedures to decide the order quantity are the same as the EOQ model for the case of increment discount, infinite supply rate, no shortage [107]. The differences are the methods to calculate the order quantity $Q^{(i)}$ associated with each of discount break point and the relative total cost.

2.1 Finite Supply Rate and No Shortage Allowed (Q6)

If replenishment rate is finite, the cost components are computed as follows:

$$\text{Purchasing cost} = DU_p^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - U_p^{(i)} q_i)$$

$$\text{Holding cost} = \frac{Q^{(i)}}{2} I(1 - \frac{D}{R_s})(\frac{C(q_i)}{Q^{(i)}} + U_p^{(i)} - U_p^{(i)} \frac{q_i}{Q^{(i)}})$$

To combine these cost components, the total annual cost function for the model is

$$TC(Q^{(i)}) = DU_p^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - q_iU_p^{(i)}) + \frac{Q^{(i)}}{2}I(1-\frac{D}{R_s})(\frac{C(q_i)}{Q^{(i)}} + U_p^{(i)} - U_p^{(i)}\frac{q_i}{Q^{(i)}}).$$

To optimise order quantity, taking derivative of the equation with respect to $Q^{(i)}$, that is

$$\frac{d\,TC(Q^{(i)})}{d\,Q^{(i)}} = -\frac{D}{Q^{(i)}}(C_p + C(q_i) - q_iU_p^{(i)}) + \frac{U_p^{(i)}I}{2}(1-\frac{D}{R_s})$$

Setting the resulting equation to zero and solving for $Q^{(i)}$, we have the order quantity for every discount break point

$$Q^{(i)} = \sqrt{2D(C_p + C(q_i) - q_iU_p^{(i)})/(U_p^{(i)}I(1-D/R_s))}.$$

2.2 Infinite Supply Rate and Shortage allowed (Q9)

To simplify the model's derivation procedure, the holding cost $C_h^{(i)}$ is estimated by

$$C_h^{(i)} = I\cdot(\sum_{j=0}^{i} U_p^{(j)})/(i+1).$$

Then the cost components, which holds the condition $q_i < Q^{(i)} < q_{i+1}$, can be computed as follows:

$$\text{Purchasing cost} = DU_p^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - q_iU_p^{(i)})$$

$$\text{Holding Cost} = \frac{C_h^{(i)}}{2}\frac{(Q^{(i)} - B^{(i)})^2}{Q^{(i)}}$$

$$\text{Shortage Cost} = \frac{C_sB^{(i)2}}{2Q^{(i)}}$$

To combine these cost components, the total annual cost function for the model is

$$TC(Q^{(i)},B^{(i)}) = DU_p{}^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - q_iU_p^{(i)}) + \frac{C_h^{(i)}}{2}\frac{(Q^{(i)} - B^{(i)})^2}{Q^{(i)}} + \frac{C_sB^{(i)2}}{2Q^{(i)}}$$

Taking the partial derivative of the equation with respect to $Q^{(i)}$ and $B^{(i)}$, we have

$$\frac{\partial\, TC(Q^{(i)},B^{(i)})}{\partial\, \mathrm{Q}^{(i)}} = -\frac{D}{(Q^{(i)})^2}(C_p + C(q_i) - q_iU_p^{(i)}) + \frac{C_h^{(i)}}{2}(1 - \frac{B^{(i)2}}{Q^{(i)2}}) - \frac{C_sB^{(i)2}}{2(Q^{(i)})^2}$$

and

$$\frac{\partial\, TC(Q^{(i)},B^{(i)})}{\partial\, B^{(i)}} = -C_h^{(i)} + \frac{B^{(i)}C_h^{(i)}}{Q^{(i)}} + \frac{C_sB^{(i)}}{Q^{(i)}}$$

Setting each equation to zero and solving for $Q^{(i)}$, we have

$$Q^{(i)} = \sqrt{\frac{2D(C_p + C(q_i) - U_p^{(i)}q_i)(C_h^{(i)} + C_s)}{C_h^{(i)}C_s}}$$

and

$$B^{(i)} = \frac{C_h^{(i)}}{C_h^{(i)} + C_S}Q^{(i)}.$$

2.3 Finite Supply Rate and Shortage Allowed (Q12)

In this case, the holding cost $C_h^{(i)}$ is also estimated by

$$C_h^{(i)} = I\cdot(\sum_{j=0}^{i} U_p^{(j)})/(i+1).$$

Then the cost components are computed as follows:

$$\text{Purchasing cost} = DU_p{}^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - q_iU_p^{(i)})$$

$$\text{Holding Cost} = \frac{C_h^{(i)}}{2}(Q^{(i)}(1 - \frac{D}{R_s}) - B^{(i)})(1 - \frac{B^{(i)}}{Q^{(i)}} - \frac{B^{(i)}D}{Q^{(i)}(R_s - D)})$$

$$\text{Shortage Cost} = \frac{C_sB^{(i)2}}{2Q^{(i)}}(\frac{R_s}{R_s - D})$$

To combine these cost components, the total annual cost function for the model is

$$TC(Q^{(i)},B^{(i)}) = DU_p^{(i)} + \frac{D}{Q^{(i)}}(C_p + C(q_i) - q_i U_p^{(i)}) + \frac{C_s B^{(i)2}}{2Q^{(i)}}(\frac{R_s}{R_s - D})$$
$$+\frac{C_h^{(i)}}{2}(Q^{(i)}(1-\frac{D}{R_s}) - B^{(i)})(1-\frac{B^{(i)}}{Q^{(i)}} - \frac{B^{(i)}D}{Q^{(i)}(R_s - D)})$$

Taking the partial derivative of the equation with respect to $Q^{(i)}$ and $B^{(i)}$, we have

$$\frac{\partial\, TC(Q^{(i)},B^{(i)})}{\partial\, \mathrm{Q}^{(i)}} = -\frac{D}{(Q^{(i)})^2}(C_p + C(q_i) - q_i U_p^{(i)}) - \frac{B^{(i)2} C_s R_s}{2Q^{(i)2}(R_s - D)} + \frac{C_h^{(i)}}{2}(\frac{R_s - D}{R_s} - \frac{B^{(i)2}}{Q^{(i)2}} - \frac{B^{(i)2} D}{Q^{(i)2}(R_s - D)})$$

and

$$\frac{\partial\, TC(Q^{(i)},B^{(i)})}{\partial\, B^{(i)}} = \frac{B^{(i)} C_s R_s}{Q^{(i)}(R_s - D)} + \frac{C_h^{(i)}}{2}(-2 + \frac{2B^{(i)}}{Q^{(i)}} + \frac{2B^{(i)}D}{Q^{(i)}(R_s - D)})$$

Setting each equation to zero and solving for $Q^{(i)}$, we have

$$Q^{(i)} = \sqrt{\frac{2D(C_p + C(q_i) - C_i q_i)}{C_h^{(i)}} \cdot \frac{R_s}{R_s - D} \cdot \frac{C_s + C_h}{C_s}},$$

and

$$B^{(i)} = \frac{C_h(R_s - D)}{R_s(C_s + C_h)} Q^{(i)}.$$

REFERENCES

1. Russell, R. S. and Taylor III, B. W. 1995. Production and Operations Management. Prentice-Hall, Inc.

2. Johson R. A., Newell, W. T. and Vergin, R. C. 1974. Production and Operations Management. Houghton Mifflin Company.

3. Evans, J. R., Anderson, D. R., Sweeney, D. J., and Williams, T. A. 1990. Applied Production and Operations Management. West Publishing Company.

4. Orlicky, J. 1975. Material Requirements Planning: The New Way of Life in Production and Inventory Management. McGraw-Hall Book Company.

5. Plossl, G. W. 1985. Production and Inventory Control: Principles and Techniques. Prentice-Hall, Inc.

6. Schroeder, G. R. 1989. Operations Management: Decision Making in the Operations Function. McGraw-Hill Book Company.

7. Adam, E. E. and Ebert, R. J. 1982. Production and Operation Management: Concepts, Models and Behaviour. Prentice-Hall, Inc.

8. Rhodes, P. Oct. 1981. Carrying Cost May Be Less Than You've Been Told. Production & Inventory Management Review and APICS News. pp. 35-36.

9. Reimans, A., Dean, B. V., Salvador, M. S., and Ora, M. 1972. Industrial Inventory Control, Gordon & Breach, New York.

10. Fleming, I. 1992. Stock Control. OR Insight, Vol. 5, No. 4, pp. 9-11.

11. Silver E. A. 1981. Operations Research in Inventory Management: A Review and Critique. Operations Research, Vol. 29, No. 4, pp. 628-645.

12. Zanakis, S. H., Austin, L. M., Nowading, D. C. and Silver, E. A. 1980. From Teaching to Implementing Inventory Management: Problems of Translation. Interface, Vol. 10, pp. 103-110.

13. Keller, R. 1987. Expert System Technology: Development and Application. Yowrdon Press.

14. Iglehart, D. L. 1967. Recent Results in Inventory Theory. The Journal of Industrial Engineering, pp. 48-51.

15. Aggarwal, S. C. 1974. A Review of Current Inventory Theory and Its Applications. International Journal of Production Research, Vol. 12, No. 4, pp 442 - 482.

16. Harris, F. W. 1915. Operations and Cost (Factory Management Series). A. W. Shaw Company. Chapter 2. pp. 48-52.

17. Raymond, F. E. 1931. Quantity and Economy in Manufacture. McGraw-Hill Book Company, New York.

18. Mykytka, E. F. and Ramberg, J. S. 1984. On the Sensitivity of the EOQ to Errors in the Forecast of Demand. IIE Transaction. Vol. 16, No. 2, pp 144 - 151.

19. Higle, J. L. 1989. A Note on the Sensitivity of the EOQ. IIE Transaction, Vol. 21, No. 3, pp. 294-297.

20. Lau, H. and Zaki, A. 1982. The Sensitivity of Inventory Decision to the Shape of Lead Time Demand Distribution. IIE Transaction, Vol. 14, No. 4, pp. 265-271.

21. Whitin, T. M. 1953. The Theory of Inventory Management. Princeton University Press, Princeton, N. J.

22. Bellman, R. E., Glicksberg, I., and Gross, O. 1955. On the Optimal Inventory Equation. Management Science, Vol. 2, pp. 83-104.

23. Arrow, K. J., Harris, T. and Marschak, J. 1951. Optimal Inventory Policy. Econometrica, Vol. 19, pp. 250-272.

24. Dvoretzky, A., Kiefer, J. and Wolfowitz, J. The Inventory Problem: I, Case of Known Distributions of Demand; Il, Case of Unknown Distributions of Demand. Econometrica, Vol. 20 (1952), pp. 187-222 and 450-466.

25. Arrow, K. J., Karlin, S. and Scarf, H. 1958. Studies in the Mathematical Theory of Inventory and Production. Stanford University Press.

26. Bellman, R. E. 1957. Dynamic Programming. Princeton University Press.

27. Wagner, H. M. and Whitin, T. M. 1958. Dynamic Version of the Economic Lot Size Formula. Management Science, Vol. 5, pp. 89-96.

28. Zangwill, W. I. 1966. A Deterministic Multi-Period Production Scheduling Model with Backlogging. Management Science, Vol. 13, pp. 105-119.

29. Clark, A. and Scarf, H. 1960. Optimal Policies for a Multi-Echelon Inventory Problem. Management Science, Vol. 6, pp. 475-490.

30. Bessler, S. A. and Veinott, A. F. 1966. Optimal Policy for a Dynamic Multi-Echelon Inventory Problem. Naval Research Logistics Quarterly, Vol. 13, pp. 355-389.

31. Willis, J. R. 1987. Computer Models for Business Decision. John Wiley & Sons Ltd.

32. Wagner, H. M. 1974. The Design of Production and Inventory Systems for Multi-facility and Multi-warehouse Companies. Operational Research, Vol. 22, pp. 278-291.

33. Veinott, A. F. 1968. The Status of Mathematical Inventory Theory. Management Science, Vol. 12, pp. 745-777.

34. Packer, A. H. 1967. Simulation and Adaptive Forecasting as Applied to Inventory Control. Operations Research, Vol. 15, pp. 660-679.

35. Eillon, S. and Elmaleh, J. 1970. Adaptive Limits in Inventory Control. Management Science, Vol. 16, pp. 533-548.

36. Peterson, R. and Silver, E. A. 1979. Decision Systems for Inventory Management and Production Planning. John Wiley & Sons Ltd.

37. Tinarelli, G. U. 1983. Inventory Control: Models and Problems. European Journal of Operational Research, Vol. 14, p. 1-12.

38. Kendall, D. G. 1951. Some Problems in the Theory of Queues. Journal of the Royal Statistical Society, Vol. 13, pp. 151-185.

39. Naddor, E. 1966. Inventory Systems. John Wiley & Sons Ltd. New York.

40. Hollier, R. H. and Vrat, P. 1977. A Proposal for Classification of Inventory Systems. OMEGA, Vol. 6, No. 3, pp. 277-279.

41. Barancsi, É., Bánki, G., Borlói, R., Chikán, A., Kelle, P., Kulcsár, T., and Meszéna, G. 1983. A Report of Research on Inventory Models. Engineering Costs and Production Economics, Vol. 7, p. 127-136.

42. IBM. 1967. Basic Principles of Wholesale IMPACT: Inventory Management Program and Control Techniques. E20-8105-1, IBM, White Plains, New York.

43. ICL. 1970. SCAN System 3 for Inventory Management-1900 Series. International Computers Limited, ICL House, Putney, London.

44. SIEMENS. 1970. Economic Inventory Control with HOREST, Characteristics and Special Features. Sienmens Aktiengesellschaft, Bereich DAtenverarbeitung, D-8000 Munchen 70.

45. Kleijnen, J. P. C. First Quarter 1978. Impact Revisited: A Critical Analysis of IBM's Inventory Package "IMPACT". Production and Inventory Management. pp. 71-90.

46. Kiran, A. S. and Loewenthal, A. 1985. An Integrated Decision System for Inventory Management, Computers and Industrial Engineering, Vol. 9, No. 4, pp. 379-386.

47. Moras, R. G. 1989. MICROLOT: Inventory Control System on the Microcomputer. Computers and Industrial Engineering, Vol. 17, No. 1-4, pp. 441-446.

48. Hill, A. V. and Mabert, V. A. 1989. A Decision Support System for Determining Optimal Retention Stocks for Service Parts Inventories. IIE Transactions, Vol. 21, No. 3, pp. 221-229.

49. Ginzberg, M. J. and Stohr, E. A. 1982. Decision Support Systems: Issue and Perspectives. in Decision Support Systems, edited by GInzberg, M. J., Reitman, W. R. and Stohr, E. A. 1982, North-Holland, Amsterdam.

50. Keen, P. G. W. and Scott Morton, M. S. 1978. Decision Support Systems: An Organisational Perspective, Addison-Wesley, Reading, Massachusetts.

51. Sol, H. G. 1985. DSS: Buzzword or OR Challenge. European Journal of Operational Research, Vol. 22, No. 1 pp. 1-8.

52. Scott Morton, M. S. 1971. Management Decision Systems: Computer-Based Support for Decision Making, Harvard University, Boston.

53. Keen, P. G. W. 1980, Spring. Decision Support Systems: Translating Analytic Techniques into Useful Tools, Sloan Management Review, pp. 33-44.

54. Gorry, G. A. and Scott Morton, M. S. 1971. A Framework for MIS. Sloan Management Review, Vol. 13, No. 1. pp. 55-70.

55. Anthony, R. 1965. Planning and Control System: A Framework for Analysis. Graduate School of Business Administration, Harvard University, Boston.

56. Simon, H. A. 1977. The New Science of Management Decision. Prentice-Hall, Englewood Cliffs, New Jersey.

57. Keen, P. G. W. 1980. Decision Support Systems: A Research Perspective in DSS: Issues and Challenge edited by Fick, G. and Sprague, R. H. 1980. Pergamon, Oxford.

58. McCosh, A. M. and Scott Morton, M. S. 1978. Management Decision Support Systems, Macmillan, London.

59. Sprague, R. H. and Carlson, E. D. 1982. Building effective Decision Support Systems. Prentice-Hall, Englewood Cliffs, New Jersey.

60. EOM, H. B. and Lee, S. M. 1990. A Survey of Decision Support System Applications. Interface, Vol. 20, No. 3, pp. 65-79.

61. Turban, E. and Trippi, R. 1989. The Utilization of ES in OR/MS: An Assessment. Omega, Vol. 17, No. 4, pp. 311-322.

62. Turban, E. and Watkins, P. R. 1986. Intelligent Expert Systems and Decision Support Systems. MIS Quarterly, Vol. 10, pp. 121-136.

63. Wynne, B. 1984. A Domination Sequence - MS/OR, DSS, and the 5th Generation. Interface, Vol. 14, No. 3, pp. 51-58.

64. Naylor, C. 1987. Building Your Own Expert System. Sigma Press.

65. Feigenbaum, E. A. 1979. Themes and Case Studies of Knowledge Engineering in Expert Systems in the Microelectronics Age, edited by Michie, D. pp. 3-25. Edinburgh University.

66. Feigenbaum, E. A. 1983. Knowledge Engineering: The Applied Side in Intelligent Systems: The Unprecedented Opportunity. pp. 37-55. Haksted Press, Wiley, New York.

67. Giarratano, J. and Riley, G. 1989. Expert Systems: Principles and Programming. PWS-KENT Publishing Company.

68. Gisolfi, A. and Balzano, W. 1993. Constructing and Consulting the Knowledge Base of an Expert Systems Shell. Expert Systems, Vol. 10, No. 1, pp. 29-34.

69. Stefik M. and Aikins, J. 1982. The Organisation of Expert Systems: A Tutorial. Artificial Intelligence Vol. 18, p.135-173.

70. Buchanan, B. G. and Shortliffe, E. H. 1984. Rule-Based Expert Systems. Addision-Wesley Publishing Company.

71. Pigford, D. V. and Baur, G. 1990. Expert Systems for Business: Concepts and Applications. Boyd & Fraser Publishing Company.

72. Hendler, J. A. 1988. Expert Systems: The User Interface. Ablex Publishing Corporation.

73. Gradwick, P. D. G. 1991. Designing a Practical User Interface for an Expert System Shell. Expert Systems, Vol. 8, No. 4, pp. 273-276.

74. McGraw, K. L. 1992. Designing and Evaluation User Interfaces for Knowledge-Based Systems. Ellis Horwood Ltd.

75. Drenth, H. and Morris, A. 1992. Prototyping Expert Solutions: An Evaluation of Crystal, Leonardo, GURU, and ART-IM. Expert Systems, Vol. 9, No. 1, pp. 35-45.

76. KBS Development Tools. August 1993. Expert System, Vol. 10, No. 3, pp. 199-202.

77. KBS Development Tools. February 1995. Expert System, Vol. 12, No. 1, pp. 75-77.

78. Jackson P. 1990. Introduction to Expert Systems (2nd edition). Addiso-Wesley Publishing Company.

79. DIT. 1992. Knowledge-Based Systems Survey of UK Applications. Department of Trade & Industry, UK.

80. Waterman, D. A. 1985. A Guide to Expert Systems. Addiso-Wesley Publishing Company.

81. Duda, R. O. and Reboh, R. 1983. AI and Decision Making: The PROSPECTOR Experience in Artificial Intelligence Applications for Business. Ablex Publishing Corporation.

82. Buchant, J, and Mcdermott, J. R. Fall 1984. R1 Revisited: Four Years in the Trenches. AI Magazine, Vol. 3, pp. 21-32.

83. Kastner, J. K. 1984. A Review of Expert Systems. European Journal of Operational Research, Vol. 18, pp. 285-292.

84. Assad, A. A. and Golden, B. L. 1986. Expert Systems, Microcomputers, and Operations Research. Computer & Operations Research, Vol. 13, Nos. 2/3, pp 301-321.

85. Messier, W. F. and Hansen, J. V. 1984. Expert Systems in Accounting and Auditing: A Framework and Review in Decision Making and Accounting: Current Research. University of Oklahoma Press.

86. Srinivasan, K., Dastoor, H. P. and Jayaraman, S. 1992. FDAS: Architecture and Implementation. Expert Systems, Vol. 9, No. 3, p. 115-124.

87. Masud, A. S. M. and Thenappan, M. S. 1993. A Knowledge-Based Advisory System for Statistical Quality Control. International Journal of Production Research, Vol. 31, No. 8, pp. 1891-1900.

88. Lavalle, J. C., Collantes, R., Sanz, M. A., and Palacios, R. 1993. SEDIMAHE: An Expert System to Help the Maintenance of Machine Tool. Maintenance, Vol. 8, No. 3, pp. 10-14.

89. Collopy, F. and Armstrong, J. S. 1992. Rule-Based Forecasting: Development and Validation of an Expert Systems Approach to Combining Time Series Extrapolation. Management Science, Vol. 38, No. 10, pp. 1394-1414.

90. Leinweber, D. 1988. Knowledge-Based System for Financial Applications. IEEE Expert, Vol. 3, pp. 18-31.

91. Kobbacy, K. A. H., Proudlove, N. C., and Harper, M. A. 1995. Towards an Intelligent Maintenance Optimisation System. Journal of the Operational Research Society, Vol. 46, pp. 831-853.

92. Grogono, P. and Baterekh, A. 1991. Expert System Evaluation Techniques: A Selected Bibliography. Expert Systems, Vol. 8, No. 4, pp. 227-238.

93. Vadera Sunil. 1989. Expert System Applications. Sigma Press.

94. Prerau, D. S. 1990. Developing and Managing Expert Systems: Proven Techniques for Business and Industry. Addiso-Wesley publishing Company.

95. Giarratano, J. and Riley, G. 1989. Expert Systems: Principles and Programming. PWS-KENT Publishing Company.

96. Newell, A. 1982. The Knowledge Level. Artificial Intelligence, Vol. 18, pp. 87-127.

97. Guida, G. and Zanella, M. 1993. Knowledge-Based Design Using the Multi-Modelling Approach. pp. 174-208. In Second Generation Expert Systems, edited by David Jean-Marc, Krivine Jean-Paul. Springer-Verlag Berlin Heidelberg.

98. David, J. M., Krivine, J. P., and Simmons, R. 1993. Second Generation Expert Systems: A Step Forward in Knowledge Engineering. In Second Generation Expert Systems, edited by David J. M., Krivine J. P., and Springer, V., Berlin Heidelberg.

99. Aamodt, A. and Plaza, E. 1994. Case-Based Reasoning: Foundation Issues, Methodological Variations and System Approaches. AI Communications, Vol. 7, No. 1, pp. 39-59.

100. Watson, I. D. and Abdullah, S. 1994. Developing Case-Based Reasoning Systems: A Case Study in Diagnosing Building Defects. In IEE Colloquium on Case-Based Reasoning: Prospects for Applications, pp. 1/1-1/3.

101. Yang, S. and Robertson, D. 1994. A Case-Based Reasoning System for Regulatory Information. In IEE Colloquium on Case-Based Reasoning: Prospects for Applications. pp. 31-33.

102. Gallant, S. I. 1993. Neural Network Learning and Expert Systems. The MIT Press.

103. Nikolopoulos, C. and Fellrath, P. 1994. A Hybrid Expert System for Investment Advising. Expert Systems, vol. 11, No. 4, pp. 245-249.

104. Hollingum, J. 1990. Expert Systems: Commercial Exploitation of Artificial Intelligence. IFS Ltd.

105. Watson, H. J. 1990. The Evoluation Continuee, in DSS-90 Transactions edited by Volonino, L. 1990. TIMS College on Information Systems, College on Information Systems, pp. 1-3.

106. Phelps, R. I. 1981. Replacement Policies under Minimal Repare. Journal of the Operational Research Society, Vol. 32, No. 7, pp. 13-20.

107. O'Keefe, R. M. 1988. Artificial Intelligent and the Management Science Practioner: Expert Systems and MS/OR Methodology Good News and Bad. Interface, Vol. 18, No. 6, pp. 105-113.

108. Dewhurst, F. W. and Gwinnet, E. A. 1990. Artificial Intelligent and Decision Analysis. Journal of the Operational Research Society, Vol. 41, No. 8, pp. 693-701.

109. Turban, E. 1993. Decision Support and Expert System: Management Support Systems, Macmillan, New York.

110. Hosseini, J., Baharaeen S., and Zheng X. T. Spring 1988. Design of a Knowledge-Based System for Inventory Control. Journal of Engineering Computing and Application, pp. 28-35.

111. Sinha D., Ghiaseddin, N., and Matta, K. 1989. Expert Systems for Inventory Control Management. Computers and Industrial Engineering, Vol. 17, Nos. 1-4, pp. 425-429.

112. Parlar, M. 1989. EXPIM: A Knowledge-Based Expert System for Production/Inventory Modelling. International Journal of Production Research, Vol. 27, No. 1, pp. 101-118.

113. Luxhoj, J. T., Agnihotri, D., Kazunas, S., and Nambiar, S. 1993. A Proposed Knowledge-Based System (KBS) for Selection of Inventory Control Policies. International Journal of Production Research, Vol. 31, No. 7, pp. 1709-1720.

114. Bramer, M. A. 1988. Expert Systems in Business: A British Perspective. Expert Systems, Vol. 5, No. 2, pp. 104-117.

115. Land, L. and Hickman, F. 1993. Advanced Technologies Survey in the UK Financial Sector. Expert Systems, Vol. 10, No. 2, pp. 103-110.

116. Buchanan, B. G. 1986. Expert Systems: Working Systems and the Research Literature. Expert Systems, Vol. 3, No. 1, pp. 32-51.

117. Perry, W. E. 1987. Effective Methods of EDP Quality Assurance. Wellesley, QED Information Science.

118. Gorry, G. A. and Scott Moton, M. S. 1971. A Framework for MIS. Sloan Management Review, Vol. 13, No. 1, pp. 55-70.

119. Bennet, J. L. 1983. Building Decision Support Systems. Addison-Wesley Publishing Company.

120. Feigenbaum, A. E. 1981. Handbook of Artificial Intelligence. Heuris Tech Press/William Kaufman, Inc.

121. Farnum, N. R. and Stanton, L. W. 1989. Quantitative Forecasting Methods. PWS-KENT Publishing Company.

122. Conover, W. J. 1971. Practical Nonparametric Statistics. Jon Wiley & Sons Inc.

123. Lewise, C. D. 1970. Scientific Inventory Control. Butterworth Publishing Ltd.

124. Muir, J. W. 1980. Forecasting Items with Irregular Demand. American Production and Inventory Society Conference Proceedings, 23rd. Annual International Conference. pp. 143-145.

125. Hays, W. L. 1981. Statistics (third edition), CBS College Publishing.

126. Press, W. H., Flannery, B. P., Teukolsky, S. A, and Vetterling, W. T. 1986. Numerical Recipes: the art of scientific computing. Cambridge University Press.

127. Lingren, B. W. 1976. Statistical Theory. Macmillan Publishing Co., Inc.

128. Kendall, M. and Stuart, A. 1977. The Advanced Theory of Statistics: Inference and Relationship, Vol. 2. Charles Griffin & Company Ltd.

129. Wang, W. 1996. Subjective Estimation of the Delay Time Distribution in Maintenance Modelling. Research Report. Department of Maths and Computer Science, The University of Salford.

130. Hadley, G. and Whitin, T. 1963. Analysis of Inventory System, Prentice Hall, Eaglewood Cliffs, New Jersey.

131. Eilon, S and Elmaleh, J. 1968. An Evaluation of Alternative Inventory Control Polices. The International Journal of Production Research, Vol. 7, No. 1, pp. 3-14.

132. Archibald, B. C. and Silver, E. A. 1978. (s, S) Polices under Continuous Review and Discrete Compound Poisson Demand. Management Science. Vol. 24, No. 9, pp. 889-909.

133. Krakewski, L. J. and Ritzman, L. P. 1990. Operations Management Strategy and Analysis. Addison-Wesley Publishing Company.

134. Sani, B. and Kingsman, B. G. 1995. Selecting the Best Periodic Inventory Control and Demand Forecasting Methods for Low Demand Items. Research Report, Management School, the University of Lancaster, UK.

135. Diegel, A. 1966. A Linear Approach to the Dynamic Inventory Problem. Management Science, Vol. 12, pp. 530 - 540.

136. Donaldson, W. A., 1977. Inventory Replenishment Policy for a Llinear Trend in Demand: An Analytical Solution. Operational Research Quarterly. Vol. 28, pp. 663 - 670.

137. Louis, J. and Brosseau, A. 1982. An Inventory Replenishment Policy for the Case of a Linear Decreasing Trend Demand. INFO, vol. 20, No. 3, pp 252-257.

138. Phelps, R. I. 1980. Optimal Inventory Rule for a Linear Trend in Demand with a Constant Replenishment Period. Journal of Operational Research Society, vol. 31, pp. 439-422.

139. Ritchie, E. 1984. The EOQ for Linear Increasing Demand: A Simple Optimal Solution. Journal of Operational Research Society, Vol. 35, pp. 949 - 952.

140. Ritchie, E. and Tsado, A. 1985. Stock Replenishment Quantities for Unbound Linear Increasing Demand: An Interesting Consequence of the Optimal Policy. Journal of Operational Research Society. Vol. 36, pp. 737 - 739.

141. Silver, E. A. 1979. A Simple Inventory Replenishment Rule for a Linear Trend in Demand. Journal of Operational Research Society, Vol. 30, pp. 71 - 75.

142. Tsado, A. 1985. Evaluation of the Performance of Lot-sizing Techniques Under Deterministic and Stochastic Demand. Unpublished PhD. Thesis, University of Lancaster, England.

143. Goyal, S. K. and Gommes, L. F. 1982. Determination of Replenishment Intervals for a Linear Trend in Demand. Working Paper, Department of Quantitative Methods, Concordia University, Montreal, Canada.

144. Amrani, M. and Rand, G. K. 1990. An Eclectic Algorithm for Inventory Replenishment for Items with Increasing Linear Trend in Demand. Engineering Costs and Production Economics, Vol. 19, pp. 261 - 266.

145. Mitra, A., Cox, J. F. and Jesse, R. R. 1984. A Note on Determining Order Quantities with a Linear Trend in Demand. Journal of Operational Research Society, Vol. 35, pp. 141- 144.

146. Ritchie, E. and Tsado, A. 1986. The Penalties of Using the EOQ: A Comparison of Lot-sizing Rules for Liner Increasing Demand. Production and Inventory Management, first quart, pp. 12-17.

147. Hertz, D. B. and Schaffir, K. H. 1960. A Forecasting Method for Management of Seasonal Style Goods Inventory. Operations Research, Vol. 8, pp 45-52.

148. Levary, R. R. 1979. A Sequential Inventory Control Model with Seasonal Demand, Omega, Vol. 8, No. 2, pp 243 - 247.

149. Wagner, H. M. and Whitin, T. M. 1958. Dynamic Version of Economic Lot Size Model. Management Science, Vol. 5, pp 89-96.

150. Kaimann, R. A. 1968. A Fallacy of EOQing. Production and Inventory Management, Vol. 9, No. 1, pp. 82-87.

151. Kaimann, R. A., 1969. Revisiting a Fallacy of EOQing. Production and Inventory Management, Vol. 9, No. 4, pp. 12-19.

152. Kaimann, R. A., 1969. EOQ vs. Dynamic programming--Which One to Use for Inventory Ordering. Production and Inventory Management, Vol. 10, No. 4, pp. 66-74.

153. Tuite, M. F. and Anderson, W. A. 1968. A Comparison of Lot Size Algorithm Under Fluctuating Demand Conditions. Production and Inventory Management, Vol. 9, No. 4, pp. 20-27.

154. Gleason, J. M. 1971. A Computational Variation of the Wagner-Whitin Algorithm: An Alternative to the EOQ Production and Inventory Management. Vol. 12, No. 1, pp 15-22.

155. Gorenstein, S. 1970. Some Marks on EOQ v. Wagner-Whitin. Production and Inventory Management, Vol. 11, No. 2, pp. 40-46.

156. Kicks, P. and Donaldson, W. A. 1980. Irregular Demand: Assessing a Rough and Ready Lot Size Formula. Journal of Operational Research Society, Vol. 31, No. 8, pp 725-732.

157. Silver, E. A. and Meal, H. C. 2nd Qtr. 1973. A Heuristic for Selecting Lot Size Quantities for the Case of Deterministic Time-varying Demand Rate and Discrete Opportunities for Replenishment. Production and Inventory Management, , pp 64-74.

158. KBS Development Tools. May 1995. Expert System, Vol. 12, No. 2, pp. 175-177.

159. Gould, C. & Lewis, C. 1985. Designing for Usability: Key Principles and What Designers Think. Communications of ACM, Vol. 28, No. 3, pp. 300-311.

160. Microsoft Corporation. 1993. Visual Basic Programmer's Guide.

161. Spragu, R. H. and Carlson, E. D. 1982. Building Effective Decision Support Systems. Prentice Hall, Englewood Cliffs, New Jersey.

162. Muhanna, W. A. and Pick, R. A. 1994. Meta-Modelling Concepts and Tools for Model Management: A System Approach. Management Science, Vol. 40, No. 9, pp. 1093-1123.

163. Borland International, INC. 1992. Borland C^{++}: Tools and Utilities Guide.

164. Deutsch, M. 1979. Verification and Validation. in Software Engineering (R. Jensen and C. Tonies, eds.). Prentice-Hall. pp. 329-408.

164. Boehm, B. 1981. Software Engineering Economics. Prentice-Hall.

166. Pressman, R. S. 1992. Software Engineering: A Practitioner's Approach. McGraw-Hill Book Company.

167. Myers, G. 1979. The Art of Software Tests. Wiley.

168. Muhanna, W. A. and Pick, R. A. 1994. Meta-Modelling Concepts and Tools for Model Management: A Systems Approach. Management Science, Vol. 40, No. 9, pp. 1093-1123.

169.Pratt, P. J. and Adamski, J. J. 1991 Database Systems Management and Design. Boyd & Fraser Publishing Company.

170.New, S. J., Lockett, A. G., and Boaden, R. J. 1993. Modelling Operational Capacity: A Challenge for Integrated OR/IS. IFORS 93 Conference, Lisbon.

171.Levine, D. S. 1994. Neural Networks for Knowledge Presentation and Inference. Lawrence Erlbaum Associates, Publishers.

172.Lisboa, P. G. J. 1992. Neural Networks: Current Applications. Chapman & Hall.