THE HONG KONG POLYTECHNIC UNIVERSITY

Department of Industrial and Systems Engineering

An Intelligent Object Paradigm for Monitoring and Control of Mobile-Enterprises

By

JIANG Xiaonuo

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

August, 2009

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

____________________________________ (Signed)

__________Jiang Xiaonuo________________ (Name of student)

Abstract of thesis entitled:

An object paradigm for Monitoring and Control of Mobile-Enterprise

Submitted by Jiang Xiaonuo

For the degree of DOCTOR OF PHILOSOPHY

At The Hong Kong Polytechnic University in August 2009

Abstract

In today's business world, the senior manager often has to travel widely representing the enterprise at major negotiations with their suppliers or customers on pricing and delivery time while keeping watch of his company. The Mobile-Enterprise concept has emerged as an innovation with the increasing use of PDA (Personal Digital Assistant) on wireless data networks. To fully realize the concept, simply allowing the use of PDA to search for information is incomplete. It has to be complemented with an intelligent paradigm to deal with the request for information as well as command execution.

This study proposes a Mobile-enterprise Multi-Agent Paradigm (MMAP) which is an agent-based intelligent solution provider developed for SMEs (Small to Medium Enterprises) to allow effective remote enquiry of business information and to exercise control in the factory/company. The paradigm

consists of two modules, namely, Real-Time Data Warehouse (RTDW), and Query and Control Multi-agent Module (QCMM). RTDW is responsible for manipulating various real-time data within the enterprise using technologies of data warehouse and OLAP. QCMM is the core module of MMAP, the framework of which is developed by multi-agent technology. The module comprises three kinds of agents to provide optimized decision support for query and control request from the end users. Furthermore, MMAP is driven by intelligent task oriented software agents where the distribution of intelligence is optimized by providing an optimum solution to a nonlinear 0-1 programming model through a scatter search approach. Its objective is to maximize the total completion rate of obtained information for a given task set. An illustration of the proposed model's application in an electronic appliance manufacturing enterprise is presented.

The main contributions in this study will include three aspects. First, a conceptual framework is established to support the development and validation of the Mobile-Enterprise. Secondly, the relevant agent architectures are proposed to carry out the intelligent operations of the system. Thirdly, a scatter search approach is developed to solve optimization problem for MMAP.

Publications Arising from the Thesis

List of International Journal Paper

[1] Jiang, X.N., Yung, K.L. (2006), "MMAP: An Intelligent object paradigm for the monitoring and control of Mobile-Enterprise", *Materials Science Forum*, Vol. 532-533, pp. 1168-1171.

[2] Jiang, X.N., Yung, K.L. (2011), "A New Intelligent System for Senior Executives to Maintain Remote Control of Their Company", *Expert Systems with Application*, Vol. 38, pp.736-742.

Acknowledgements

First of all, I would like to express my sincere appreciation and gratitude to my dissertation supervisor, Professor Kai Leung Yung for his considerable help, continual guidance and encouragement throughout the research period. His deep understanding and wide knowledge have broadened my view in the field about this research. I also wish to thank Prof. Ding Wei Wang, Prof. Jia Fu Tang and Prof. Yan Ning Zhang for many valuable suggestions they have provided to me.

Secondly, appreciation goes to my colleagues and friends, Dr Sandy To, Dr X. Q. Wang, Mr. S. M. Ko, and Mr. C. W. Lam for their inspiration, support and encouragement.

Thirdly, the financial assistance support by the Research Committee of the Hong Kong Polytechnic University through the award of research studentship is greatly appreciated.

Furthermore, without mentioning any names, I would like to express my thanks to my colleagues in the Department of Industrial and Systems Engineering for their thoughtful discussion and assistance.

Last but not least, I would like to thank my families who have whole heartedly supported my study over the past many years.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADS, Automated Decision Systems

AI, Artificial Intelligence

ANN, Artificial Neural Network

BAM, Business activity management

BDI, Beliefs-Desires-Intentions

BI, Business Intelligence

BPM, Business perfomıance management

CAD, Computer-Aided Design

CAM, Computer-Aided Manufacturing

CBIS, Computer-Based Information System

CBR, Case-Based Reasoning

CIM, Computer Integrated Manufacturing

CPM, Corporate performance management

CRI, Completion Rate of obtained Information

CRM, Customer Relationship Management

DCA, Data Collection Agent

DLL, Dynamically Linked Libraries

DM, Data Mining

DW, Data Warehouse

DSS, Decision Support System

DW, Data Warehouse

EA, Enjoin Agent

EAI, Enterprise Application Integration

EIS, Executive information systems

ERP, Enterprise Resource Planning

ES, Expert systems

ETL, Extraction, transformation, and load

GA, Genetic Algorithm

HMS, Holonic Manufacturing System

HTML, HyperText Markup Language

IDSS, Intelligent Decision Support System

IT, Information Technology

JIT, Just-In-Time

KBS, Knowledge-based System

KM, Knowledge Management

KMS, Knowledge management system

KQML, Knowledge and Query Manipulation Language

MAS, Multi-agent System

MDBMS, Multidimensional Database Management System

MES, Manufacturing Execution System

MIS, Management Information System

MMAP, Mobile-enterprise Multi-Agent Paradigm

MRP, Material Requirements Planning

MRPII, Manufacturing Resource Planning

MSS, Management Support System

OLAP, Online Analytical Processing

PDA, Personal Digital Assistant

QA, Query Agent

QCMM, Query and Control Multi-agent Module

RFID, Radio Frequency Identification

RTDW, Real-Time Data Warehouse

SA, Simulated Annealing

SCM, Supply Chain Management

SME, Small and Medium Enterprise

SQL, Structured Query Language

SS, Scatter Search

TS, Tabu Search

VPN, Virtual Private Network

XML, eXtensible Markup Language

Chapter 1 Introduction

1.1 Background

The business environment in which modem enterprises operate today is constantly changing, and it is becoming more and more complicated (Huber, 2004). Customers' requirements for diverse production variety, Just-in-time (JIT) delivery, and high quality service have challenged enterprises to be innovative in the way they operate and to respond quickly to changing conditions. Enterprise managers are required to be agile and to make frequent and quick strategic, tactical, and operational decisions to achieve higher profitability and strengthen their competitive positions in the rapidly changing marketplace (Turban, Aronson *et al.*, 2006). Nowadays, a wide variety of infonnation systems for decision support are playing vital and expanding roles in daily business activities as accounting, production, inventory, marketing, or any other major functions. Management is becoming a distributed and co-operative problem-solving activity (Ray, 2000).

On the other hand, the rapid growth of wireless networking technologies and mobile computing devices such as laptops, personal digital assistants (PDAs), smart phones and tablet computers during the last decade laid the foundation for mobile management. Many modem cities around the world have built, or are planning to build city-wide wireless networks so that citizens can use a PDA to access information anywhere within the city. This trend has generated many innovations in using it to access information sources through web-based systems and control the company with information updates, collaboration efforts, and decisions. Mobile solutions are taking on greater importance in the enterprise.

In today's globally competitive marketplace, the boss of a Small and Medium Enterprise (SME) who normally assumes many roles such as the sales, marketing and procurement person has to travel widely to negotiate with customers and suppliers on pricing and delivery time while keeping watch of his company. For example, in the case of the senior executive on the move, for supporting their snap decisions while at the same time relating their corresponding commands to parties concerned for action to be taken. To make this remote monitoring and control efficiently and in a timely manner, it is possible to develop a brand new concept – "mobile enterprise" to automatically drive local continuous improvements and other operational activities such as scheduling and planning. With this concept, corporations can provide managers with non-stop decision support, allowing them to work productively no matter where they are. Field data can be captured in real time and made available without delay. The need for this paradigm is also intensified by the recent trends in the extensive globalisation of subcontracting and sales that require many company executives to be articulated and travel widely.

A large number of tools and techniques have been developed over the years to support managerial decision making, as summarized in Table 1.1.

Artificial Intelligence (AI) methods are improving the quality of decision support and have become embedded in many applications, ranging from automated pricing optimization to intelligent policy recommendations. Intelligent agents can perform routine tasks, freeing up time that decision makers can devote to important work. Furthermore, with the tremendous progress of information technology, knowledge management becomes much easier and effective, which can assist in all the practical aspects of running a professional business (Wen, Chen *et al.*, 2008). The integration of wireless technologies, data warehouse, and knowledge management delivers an entire organization's expertise on the solution of problems anytime and anywhere. However, every type of tool has certain limitations. The recent trend of solutions for decision support is the integration of several of the tools and techniques to solve managerial problems better and faster.

Table 1.1: Computerized Tools for Decision Support

Tool Category	**Tools and Their Acronyms**
Data management	Databases and database management system (DBMS)
	Extraction, transformation, and load (ETL) systems
	Data warehouses (DW), real-time DW, and data marts
Reporting status tracking	Online analytical processing (OLAP)
	Executive information systems (EIS)
Business analytics	Optimization
	Data mining
	Web analytics
Strategy and performance management	Business performance management (BPM)/Corporate performance management (CPM)

	Business activity management (BAM)
Knowledge management	Knowledge management systems (KMS)
Intelligent systems	Expert systems (ES)
	Artificial neural networks (ANN)
	Fuzzy logic
	Genetic algorithms
	Intelligent agents
Enterprise systems	Enterprise resource planning (ERP), Customer relationship management (CRM), and supply chain management (SCM)

Although the growth of enterprise-wide information systems provides multiple functions and extraordinary rich data to managers, the software are still not easy to operate, and the requirements for complex information which is not stored in current databases and may demand more processing cannot be satisfied. During important negotiations, there is not enough time for managers to get instant access to the really important real-time information that they need to make accurate and timely decisions. In addition, it is yet a challenge for recent software to effectively split up an order/command into different tasks and issue them to the relevant part of the organization.

Therefore, this research aims to develop a Mobile Enterprise Multi-agent Paradigm (MMAP), with a combination of existing technologies, in order to provide the executives on the move with the right information and decision support functions to commit sale of their products while at the same time be able to control and manage their company operation.

1.2 Problem Statement

Therefore, the focus of this research is on the design of an effective holistic paradigm which is capable of matching the exact requirement of the executives on the move in order to increase their flexibility and competitiveness in the market. To achieve this objective, in-depth analysis should be conducted in order to generate a set of solutions for benefiting the executives in the enterprise environment.

Although there are currently a number of commercial enterprise mobility solutions available in the market developed for "mobile enterprise", most of them are focused on the integration of wireless technology with enterprises management applications (Brans, 2002; Hayes and Kuchinskas, 2003; Kornak, Teutloff *et al.*, 2004). Wireless information retrieval alone is too simple to meet advanced requirements.

In addition, almost all traditional information systems focus on simple and predefined formulas for managing manufacturing, production, sales, etc. Even e-commerce and m-commerce mainly deal with selling goods or providing services on the web. They lack of the functions of automatic knowledge reasoning and management for running a company. To remotely monitor and control of an enterprise in an effective way, the supporting systems, tools and structure must also evolve. The need to increase responsiveness to key customers and suppliers' requirements for information, services, decisions and creative collaboration has stimulated recent research effort towards the integration of AI with Decision Support Systems (DSSs), which is called Intelligent Decision Support Systems (IDSS). A review of literature shows that managerial decision support can be improved by a number of integrated approaches and systems, such as combining intelligent agents, data mining techniques with ERP systems proposed by (Symeonidis, Kehagias *et al.*, 2003; Lea, Gupta *et al.*, 2005). However, a real-time intelligent paradigm, which is not only able to support important decision, but also capable of monitoring and control the company by a mobile device, is still an area that requires more in-depth study and investigation. There are still major issues to be resolved. These are summarized as follows:

1) How to collect the relevant real-time data and information from distributed functional modules especially on the production floor to combine personnel movements with machine and inventory data, and store them using standardized data formats.
2) How to effectively accommodate all the activities required for the extraction of information from the appropriate sources, automatically collate them into an appropriate answer and present them to the requester in an easily understandable form.
3) How to execute the instructions given by automatically retrieved information, determine who in the company should take action and distribute the instructions together with the required information.
4) How to improve system efficiency and accuracy by optimization.

5) How to implement the intelligent object paradigm in real business management.

1.3 Research Objectives

The specific objectives of this research are to investigate:

1) The issues and the mechanisms involved in the collection of real-time infonnation from various departments of a company.
2) The collation of this real-time data into a database for access by other analytical and scheduling applications from different platforms.
3) The formulation of an object paradigm for smooth mapping of physical operations into logical schematics for design of the above applications.
4) The development of interfaces for wireless PDA (Personal Digital Assistant) access into the real-time and batch information within the company.

1.4 Significance of the Research

Traditional approaches are confined to focusing on the specific process optimization for recommendations or queries and are not sophisticated enough to provide insight into how management instructions can be issued to the relevant part within the organization. This research is concerned with the introduction of MMAP, aiming at supporting SME executives on the move to make snap decisions and remotely control the enterprise in an effective way. This architecture provides a novel function to issue real-time command and execute the instruction intelligently within the company. Most work in this aspect is original and makes a contribution to the optimization of the intelligence distribution strategy.

1.5 Thesis Outline

This thesis consists of six chapters.

Chapter one introduces the background of Mobile Enterprise and the motivation of the study. The problems that occur in existing enterprise-wide management information systems are stated.

Chapter two is an academic review on existing methods for decision-making adopted by various industries. Some related technologies including data acquisition and data warehousing, optimization algorithms, user interface and secure mobile service platfonn are also discussed in this chapter.

Chapter three describes the infrastructure of MMAP and presents the system architecture which consists of Real-Time Data Warehouse (RTDW) and Query and Control Multi-agent Module (QCMM).

Chapter four attempts to tackle a design and optimization decision for the MMAP. The method hinges on providing an optimized solution to a nonlinear 0-1 programming model through a scatter search approach. A mathematical model for the addressed problem is developed; its objective is to maximize the total completion rate of obtained information for the given task set.

Chapter five provides an illustration of MMAP with an application scenario and operations to affect monitoring function. The QA software prototype is also demonstrated in this chapter.

Chapter six draws the overall conclusions from the work undertaken. In this chapter the contribution made by this study is presented, and areas for future research are identified.

Chapter 2 Literature Review

2.1 Introduction

In the increasingly customer-oriented marketplace, the right decision is always the key point to achieve customer satisfaction and better collaboration. It can make the difference between success and failure. The success of a company increasingly depends on timely information being available to the right person at the right time for crucial managerial decision-making (Chen, Goes *et al.*, 2006). For example, to support and improve the monitoring and control functions, managers need information that is delivered efficiently and in a timely manner, to mobile applications embedded in PDAs and cell phones. Late and sometimes inaccurate information may lead to poor customer satisfaction and so the ultimate goal is to ensure that the right decision is always made anytime, anywhere. Making such decisions may require considerable amounts of relevant data, information, and knowledge.

Information becomes a vital resource to support the modern business process, which need to be processed quickly and available timely. However, the amount of information in a company is tremendous and increasingly complicated. The old information systems cannot meet the needs of managers to make fast decisions, evaluate large amounts of information that is stored in different locations, and collaborate (Turban, Aronson *et al.*, 2006). It is commonly accepted that advanced decision support systems (DSSs) should be used to fundamentally change the status. In the circumstances that the decision-making environments are more complex, new ideas are constantly emerging, and knowledge is getting more and more indefinite, the research of the intelligent decision support system (IDSS) is of great significance. New techniques such as, intelligent agents, knowledge management (KM), and optimization, added promised capabilities and easy access to tools, models, and data for computer-aided decision making.

As DSS evolved, some new tools started to appear under the name Business Intelligence (BI) in the mid-1990s. BI refers to a broad category of applications and technologies for gathering, providing access to, and analyzing data for the purpose of helping enterprise users make better business decisions (Biere, 2011). It includes the data warehousing (DW), online analytical processing (OLAP), data mining, querying techniques, and Business Performance Management (BPM), etc.

In addition, since the development of the Internet and Web servers and tools, there have been dramatic changes in how decision makers are supported. Today's decision support tools utilize the Web for their analysis, and they use graphical user interface that allow decision makers to flexibly, efficiently, and easily view and process data and models by using familiar Web browsers. A variety of handheld wireless devices, including mobile phones and PDAs, enable managers to access important infonnation systems and useful tools, communicate, and collaborate.

This chapter aims to provide a comprehensive survey of related research work in order to consolidate the traits of an intelligent paradigm for managerial decision support in modern enterprise operations.

2.2 Decision Support Systems (DSS) and Business Intelligence (BI)

For years, management and decision making were considered purely an art because a variety of individual styles could be used in approaching and successfully solving the same types of managerial problems (Turban, Aronson *et al.*, 2006). These styles were often based on creativity, judgment, intuition, and experience rather than on systematic quantitative methods grounded in a scientific approach. However, the intuition and trial-and-error approaches to managerial decision making are not effective because today's business environmental factors make such an evaluation process very difficult.

From traditional uses in transaction processing, computerized systems are now penetrating complex managerial areas ranging from the monitoring activities to problem analysis and solution applications (Geoffrion and Krishnan, 2001). Today's computerized systems possess a number of capabilities that have been driving the use of computerized decision support since the late 1960s, but especially since the mid-1990s. One of the major objectives of computerized decision support is to facilitate closing the gap between the current performance of an organization and its desired performance, as expressed in its mission, objectives, and goals, and the strategy to achieve them (Coveney, King *et al.*, 2003).

DSS are collections of computer technologies that support management tasks in general and decision-making in particular (Turban, Aronson *et al.*, 2006). Most decision-support methods provide for quick data queries and use models to convert the data into usable information for consideration by a decision-maker. DSS are becoming increasingly more critical in the daily operation of enterprises (Nemati, Steiger *et al.*, 2002).

As we begin the 21st century, we are seeing major changes in how manager use computerized support in making decisions. Corporations regularly develop distributed systems, intranets, and extranets that enable easy access to data stored in multiple locations, collaboration, and communication worldwide (Turban, Aronson *et al.*, 2006). Organizations can now easily deliver high-value performance-analysis applications to decision makers around the world who can make better decisions because they have more accurate information at their fingertips. The field of DSS/BI is evolving from its beginning as primarily a personal support tool and is quickly becoming a shared commodity across organizations.

2.2.1 Computerized Support for Decision Making

(Simon, 1977) described the decision-making process with a four-phase process of intelligence, design, choice and implementation, which are defined as follows:

1. Intelligence. This phase involves searching for conditions that call for decisions.
2. Design. This phase involves inventing, developing, and analyzing possible alternative courses of action (solutions).
3. Choice. This phase involves selecting a course of action from among those available.
4. Implementation. This phase involves adapting the selected course of action to the decision situation (i.e., problem solving or opportunity exploiting).

The relationships among the four phases are shown in Figure 2.1.

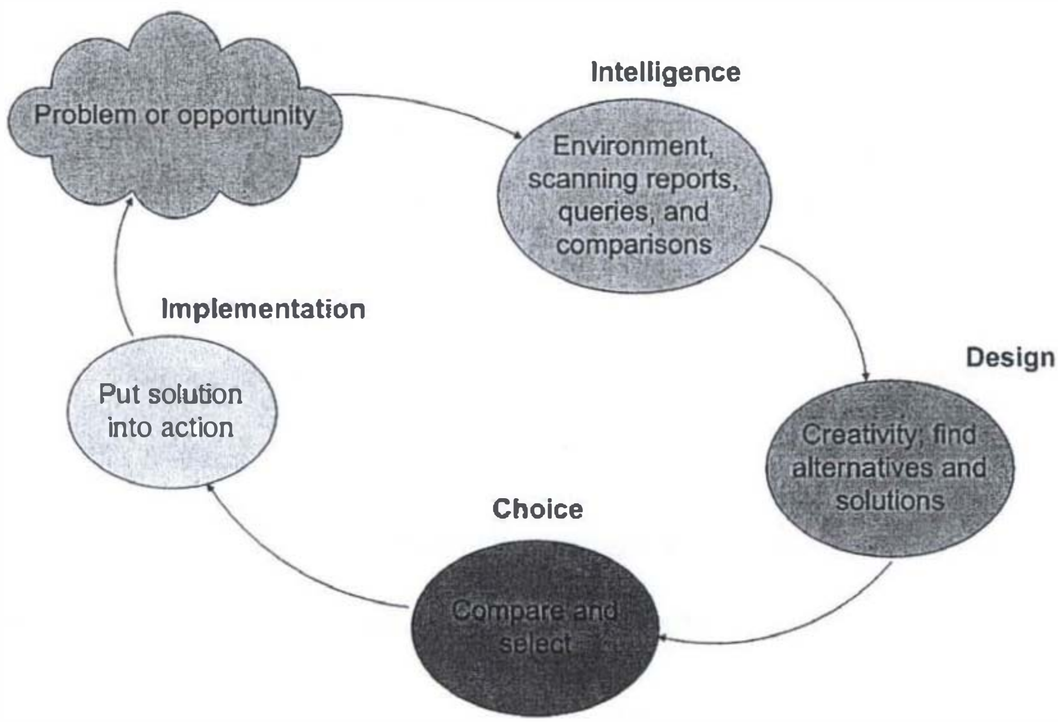

Figure 2.1 The Steps of Decision Support

According to (Simon, 1977), decisions can be classified into three types: *structured, unstructured, and semi-structured.*

2.2.1.1 Computer Support for Structured Decisions

In a structured problem, all four phases described above are structured (Simon, 1977). Structured problems that are encountered repeatedly have a high level of structure. The procedures for obtaining the best solution are known and the objectives are clearly defined. It is therefore possible to abstract, analyze, and classify them into specific categories. For each category of decision, an easy-to-apply prescribed model and solution approach called operational research (OR) have been developed, generally as quantitative formulas. The OR approach is an interdisciplinary branch of applied mathematics and formal science that uses advanced analytical methods such as mathematical modelling, statistical analysis, and mathematical optimization to arrive at optimal or near-optimal solutions to complex decision-making problems (Wilkes, 1989; Ciriani, 1999). It is often concerned with detennining the maximum (of profit, performance, or yield) or minimum (of loss, risk, or cost) of some real-world objective (Wilkes, 1989; Ciriani, 1999; Kirby, 2003).

Another approach to supporting structured decision making is automated decision systems (ADS) (Davenport and Harris, 2005). An ADS is a rule-based system that provides a solution, usually in one functional area (e.g., finance, manufacturing), to a specific repetitive managerial problem, usually in one industry (e.g., to determine the price of an item in a store). The rules, which are based on experience or derived through statistical analysis, can be combined with mathematical models to form solutions that can be automatically and instantly applied to problems, or they can be provided to a human, who will make the final decision. ADS attempt to automate highly repetitive decisions, which are mostly suitable for frontline employees who can see the customer information online and frequently must make quick decisions.

2.2.1.2 Computer Support for Unstructured Decisions

An unstructured problem is one in which none of the four phases is structured (Simon, 1977). Unstructured problems can be only partially supported by standard computerized quantitative methods. It is usually necessary to develop customized solutions, which may require certain expertise that can sometime be provided by intelligent systems, such as AI and ES (Henrik, 1996), Case-Based Reasoning (CBR) (Choy, Lee *et al.*, 2002), intelligent agents (Symeonidis, Kehagias *et al.*, 2003; Lea, Gupta *et al.*, 2005). Knowledge management (KM) may also play a large role in this type of decisions, which is the systematic and active management of ideas, information, and knowledge residing in an organization's employees (Becerra-Fernandez and Sabherwal, 2010). The structuring of knowledge enables effective and efficient problem solving, dynamic learning, strategic planning, and decision making (Becerra-Fernandez and Sabherwal, 2010; Schwartz, 2010; Szczerbicki and Nguyen, 2010). More research that discusses knowledge management techniques to support unstructured decisions can be found in (Eriksson, 1996; Matsatsinis, Doumpos *et al.*, 1997; Li, 2000; Liao, 2001; Bharati and Chaudhury, 2004).

2.2.1.3 Computer Support for Semi-Structured Decisions

Semi-structured problems fall between structured and unstructured problems, having some structured elements and some unstructured elements (Simon, 1977). Solving semi-structured problems may involve a combination of standard solution procedures and human judgment. OR can provide models for the portion of a decision-making problem that is structured. For the unstructured portion, a DSS can improve the quality of the information on which the decision is based by providing, for example not only a single solution but also a range of alternative solutions, along with their potential impacts (Symeonidis, Kehagias *et al.*, 2003). These capabilities help managers to better understand the nature of problems and thus to make better decisions.

2.2.1.4 DSS Definition

Computers have supported structured and some semi-structured decisions, especially those that involve operational and managerial control since the 1960s (Turban, Aronson *et al.*, 2006). For unstructured decisions, conventional management information systems (MIS) and OR tools might be insufficient (Burstein and Holsapple, 2008).

(Gorry and Scott-Morton, 1971) proposed the use of a supportive information system, which then evolved into a new technology – DSS. Traditional DSS was defined as "interactive computer-based systems, which help decision makers utilize data and models to solve unstructured problems". Another classic DSS definition is provided by (Keen and Scott-Morton, 1978): "Decision support systems couple the intellectual resources of individuals with the capabilities of the computer to improve the quality of decisions. It is a computer-based support system for management decision makers who deal with semi structured problems". In summary, these early definitions of a DSS identified it as a system intended to support managerial decision makers in semi-structured and unstructured decision situation. DSS were developed to be adjuncts to decision makers to extend their capabilities but not to replace their judgment.

2.2.2 The Basic Components of a DSS Architecture

DSS can be provided in many different configurations. These configurations depend on the nature of the management-decision situation and the specific technologies used for support (Turban, Aronson *et al.*, 2006). These technologies are assembled from four fundamental components – data, models, user interface, and (optionally) knowledge (Sprague and Carlson, 1982; Marakas, 1999; Power, 2002; Haag, Cummings *et al.*, 2007). Each of these components is managed by software that is either commercially available or programmed for the specific task (Power, 2009). The manner in which these components are assembled defines their major capabilities and the nature of support provided.

A typical DSS application can be composed of the data management subsystem, the model management subsystem, the user interface subsystem, and a knowledge-based management subsystem, as shown in Figure 2.2 (Turban, Aronson *et al.*, 2006).

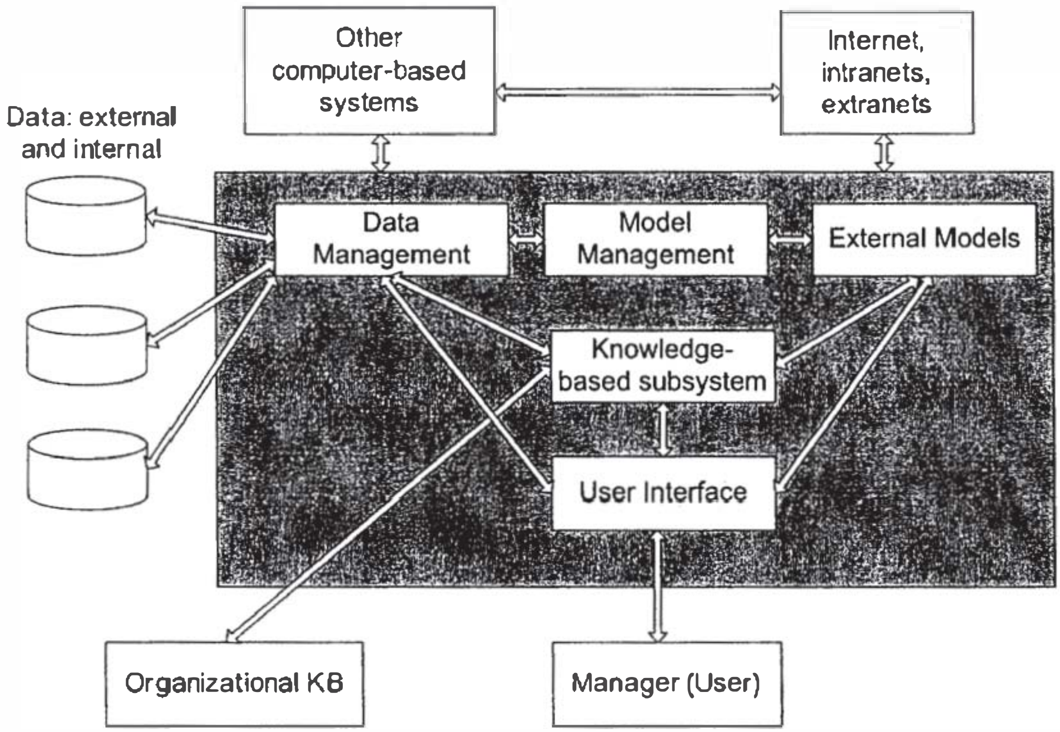

Figure 2.2 A Schematic View of DSS

2.2.2.1 The Data Management Subsystem

The DSS methodology recognizes the need for data from many sources (including the Web) to solve problems. Every problem that has to be solved and every opportunity or strategy to be analyzed requires relevant data. Therefore, data are the first component of the DSS architecture (Turban, Aronson *et al.*, 2006). DSS typically have their own databases and are developed to solve a specific problem or set of problems (Chow, Choy *et al.*, 2005).

(Turban, Aronson *et al.*, 2006) present that the data management subsystem is composed of (i) DSS database, (ii) database management system (DBMS), (iii) Data directory, (iv) Query facility, as shown schematically in

Figure 2.3. The interaction of the data management subsystem with several data sources and the other parts of the DSS is also shown in this figure.

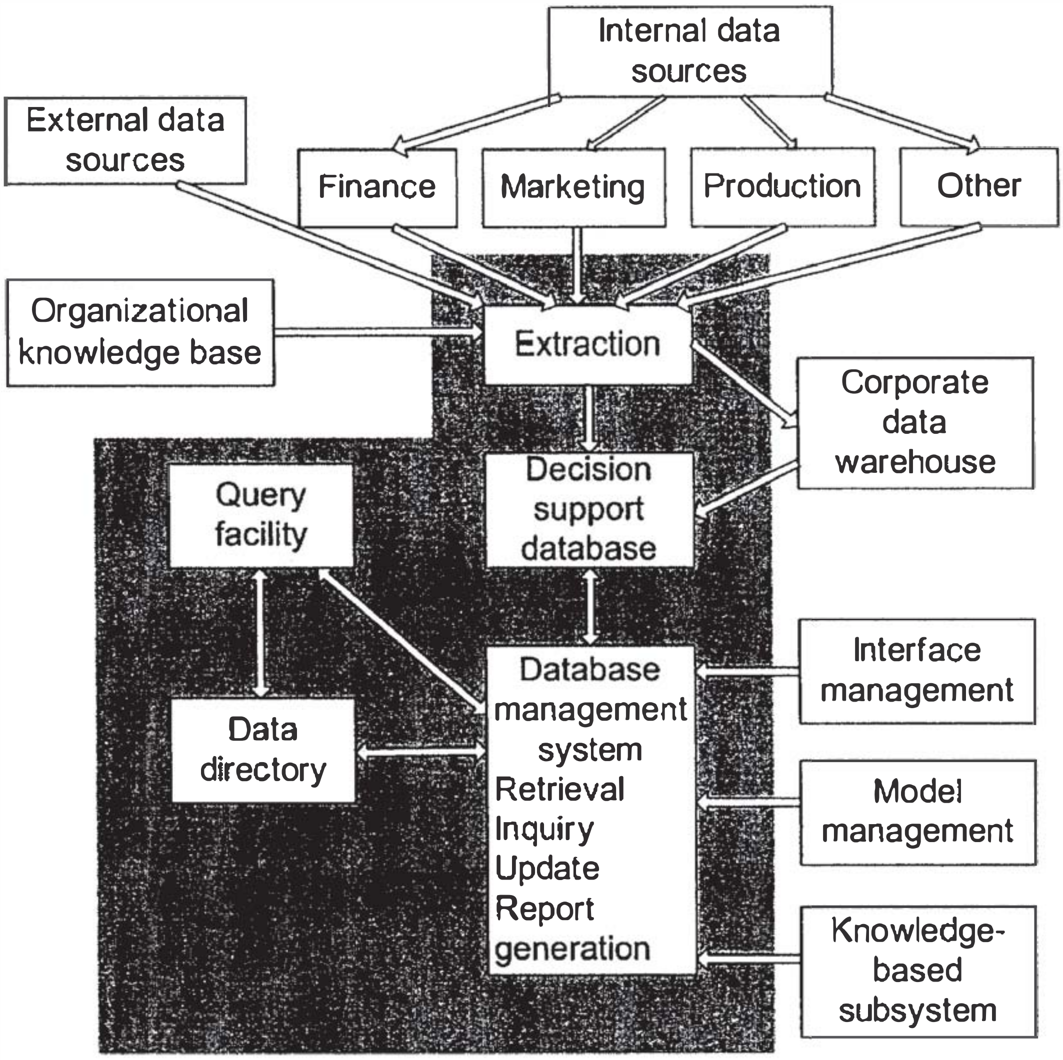

Figure 2.3 The Structure of the Data Management Subsystem

The data management subsystem may include a database that contains relevant data extracted from internal and external data sources for the situation. The extraction results will be sent to the specific application's database or to the corporate data warehouse. Both internal and external data can be maintained in the DSS database or accessed directly when the DSS is used. (Respicio, Captivo *et al.*, 2002) described a spreadsheet-oriented DSS for production planning and scheduling. This DSS database, essentially in an Excel spreadsheet, populated with data extracted from a legacy database. Object-oriented databases in Extensible Markup Language (XML) exist and are used in DSS (Ho, Lau *et al.*, 2006). XML is becoming the standard, consistent data translation method for mobile commerce devices, and it is also used for standard Web browser access to data.

A DBMS is a set of computer programs that controls the creation, maintenance, and the use of the database. Most DSS are built with a standard commercial relational DBMS that provides a number of capabilities (Cundiff, 1989). An effective database and its management can support many managerial activities (Elisa, 1991).

The data directory is a catalog of all the data in a database, which contains data definitions. Its main function is to answer questions about the availability of data items, their source, and their exact meaning (Marrs and Robinson, 1991). The query facility is built to access, manipulate, and query data. It accepts requests for data from other DSS components, determines how the requests can be filled (consulting the data directory if necessary), formulates the detailed requests, and returns the results to the issuer of the request (Conlon, Reithel *et al.*, 1994).

Although a large number of data management systems are proposed to support decision-making processes (English, 2002; Gonzales, 2004; English, 2005), data quality problems are widespread in practice and reliance on data of poor or uncertain quality leads to significant expense and less-effective decision-making (Wand and Wang, 1996; English, 1999; Redman, 2001). There is a large body of existing work on improving data quality, for example, (Redman, 1997; English, 1999; Madnick and Zhu, 2006) discuss methods of improving data quality, while (Madnick, Wang *et al.*, 2001) review current practice and research in the area. Research literature characterizing or defining data quality ranges from simple lists of quality criteria to comprehensive frameworks (Wand and Wang, 1996; English, 1999; Redman, 2001; Kahn, Strong *et al.*, 2002). (Burstein, Holsapple *et al.*, 2008) presents a semiotic-based framework that provides a rigorous and comprehensive means of understanding data quality. The framework can be used as a basis for identifying typical data quality problems, defining improvement strategies to address these problems, and defining tags to provide data quality information to decision makers.

2.2.2.2 The Model Management Subsystem

Data related to a specific situation are manipulated by using models, which are the second component of the DSS architecture. The model management subsystem of a DSS is composed of the following elements: Model base, model base management system (MBMS), Modelling language, Model directory, and Model execution, integration, and command processor (Turban, Aronson *et al*., 2006). These elements and their interfaces with other DSS components are shown in Figure 2.4.

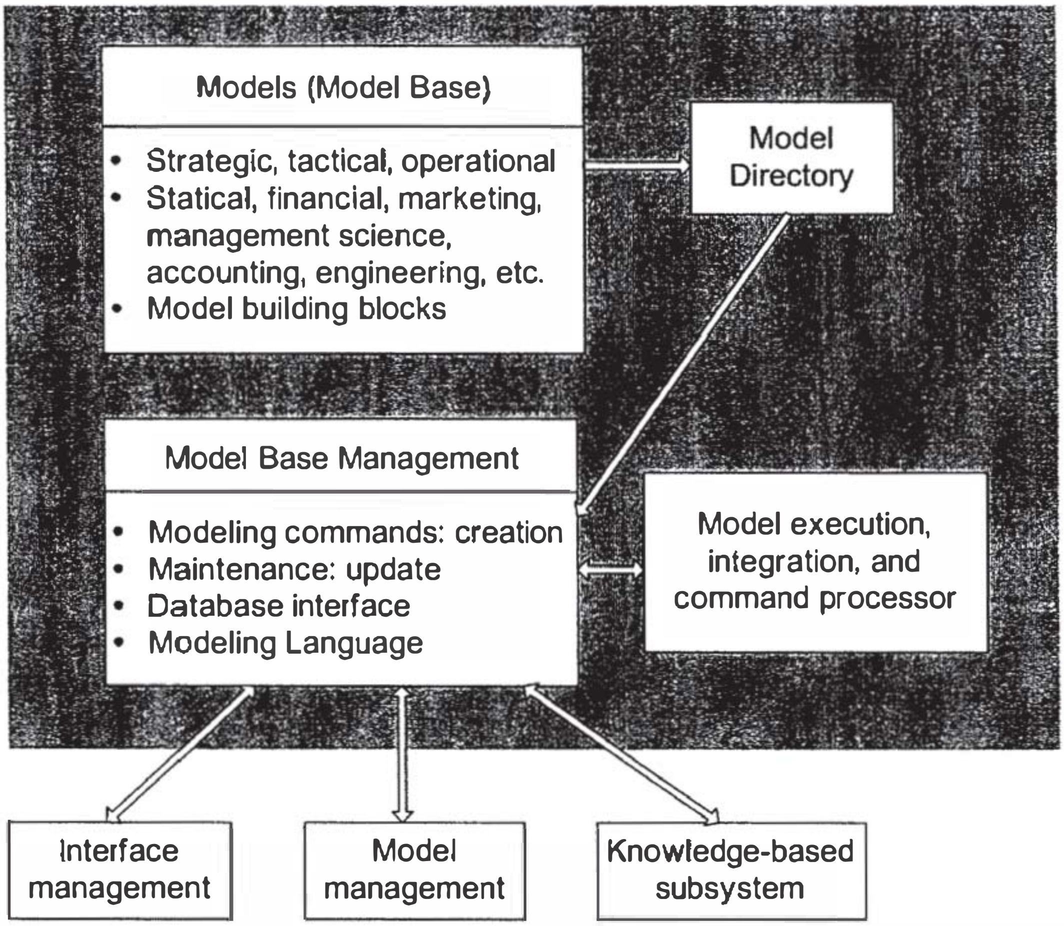

Figure 2.4 The Structure of the Model Management Subsystem

A model base contains routine and special statistical, financial, management science, and other quantitative models that provide the analysis capabilities in a DSS. Models in DSS are basically mathematical, which are expressed by formulas (Respicio, Captivo *et al.*, 2002).

MBMS software may have four main functions (Turban, Aronson *et al.*, 2006):

1. Model creation, using programming languages, DSS tools and/or subroutines, and other building blocks

2. Generation of new routines and reports

3. Model updating and changing

4. Model data manipulation

Similar to the role of a database directory, the model directory is a catalog of all the models and other software in the model base, which contains model definitions. Its main function is to answer questions about the availability and capability of the models (Song, Yu *et al*., 2009).

Model execution is the process of controlling the actual running of the model, while model integration involves combining the operations of several models when needed or integrating the DSS with other applications. A model command processor is used to accept and interpret modelling instructions from the user interface component and route them to the MBMS, model execution, or integration function (Turban, Aronson *et al*., 2006).

Many readily accessible applications describe how the models incorporated in DSS contribute in a major way to organizational success. These include Dupont (Promodel, 1999), Procter & Gamble (Camm, Chorman *et al.*, 1997), Fiat (Promodel, 2006).

2.2.2.3 The User Interface Subsystem

A user interface is the third component of the DSS architecture (Turban, Aronson *et al*., 2006)(http://en.wikipedia.org/wiki/Decision_support_system).

The user communicates with and commands the DSS through the user interface subsystem, as shown schematically in Figure 2.5. It is managed by the user interface management system (UIMS), which enables the user to interact with the model management and data management subsystems. (Rathnam and Mannino, 1995) surveys UIMS and its role in the construction of the user interface of DSSs. In advanced systems, the user interface component includes a natural language processor or can use standard objects (e.g., pull-down menus, buttons) through a GUI.

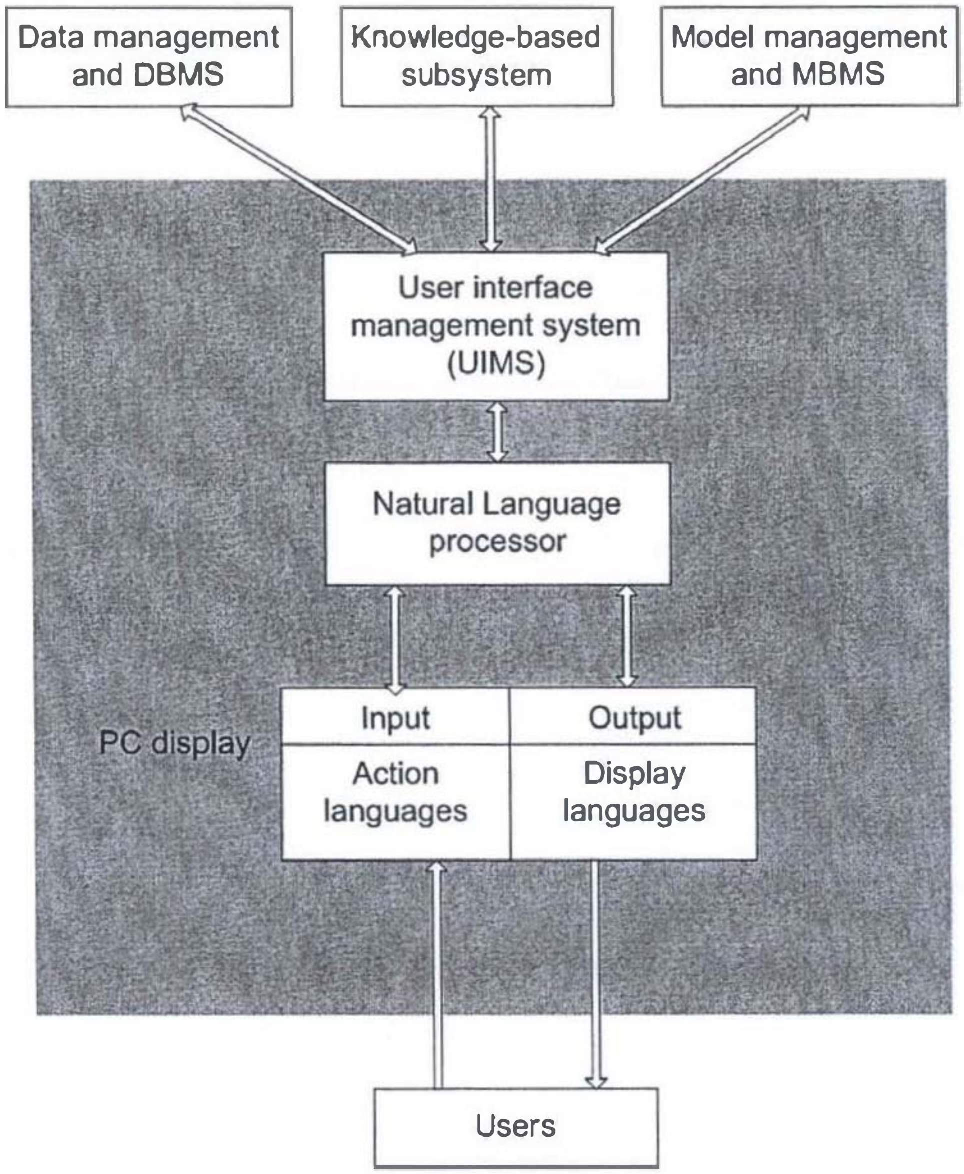

Figure 2.5 Schematic View of the User Interface System

Today's DSS utilize Web browser technology or a customized graphical user interfaces (GUI) with a similar-looking screen that allow decision makers to flexibly, efficiently, and easily view and process data and models (Xie, Wang *et al.*, 2005; Wen, Chen *et al.*, 2008). Given that Web browsers are the typical user access path for DSS and data, most major software vendors, provide portals or dashboards, which allow access to data and methods to solve problems. IBM's WebSphere portal (Ben-Natan, Sasson *et al.*, 2004) lets employees access information and business processes in a comfortable format, leading to collaboration and faster and better decision making. This includes IBM's e-commerce and SCM applications. The WebSphere portal provides capabilities that allow an organization to connect everyone involved in its supply chain together so that collaboration is smooth and effective. EIS ranging from SCM to KM can effectively be accessed though the Microsoft Digital Dashboard and IBM WebSphere portal. Web-based DSS have reduced technological barriers and made it easier and less costly to make decision-relevant information available to managers and staff user in geographically distributed locations, especially through mobile devices (Bhargava, Power *et al.*, 2007).

A variety of portable devices have been made Web-ready, including notebook and tablet PCs, PDAs, pocket PCs, and smart phones (Wen, Chen *et al.*, 2008). Many of these devices allow either handwritten input or typed input from internal or external keyboards. Some DSS user interfaces utilize natural language input so that the users can easily express themselves in a meaningful way. Because of the fuzzy nature of human language, it is fairly difficult to develop software to interpret it. However, scientists are constantly developing improvements in natural language processing and language translation (Kaufmann and Bernstein; Lim and Lee; Ribeiro and Moreira, 2003; Conlon, Conlon *et al.*, 2004; De Francesco, Santone *et al.*, 2007; Cimiano, Haase *et al.*, 2008; Tapeh and Rahgozar, 2008; Bos, 2009). Artificial intelligence methodologies and technologies directly affect these improvement efforts (Anthes, 2006).

2.2.2.4 The Knowledge-Based Management Subsystem

Many unstructured and even semi-structured problems are so complex that their solutions require expertise. Therefore, advanced DSS are equipped with a component that is a knowledge-based management subsystem which can support any of the other subsystems or act as an independent component (Turban, Aronson *et al.*, 2006). It can be interconnected with the organization's knowledge repository (part of a KMS). The knowledge component consists of one or more intelligent systems. Like database and model management software, knowledge-based management software provides the necessary execution and integration of the intelligent system. This component can supply the required expertise for solving some aspects of the problem and provide intelligence that can enhance the operation of other DSS components. A DSS that includes such a component is called an intelligent DSS, or a knowledge-based DSS. There have been many attempts to use knowledge-based DSS to solve managerial problems in modem business operations. (Zopounidis, Doumpos *et al.*, 1997) developed a knowledge-based decision support system for financial management that integrates the technologies of decision support systems and was used to tackle past and current frequently occurring problems. (Wen, Wang *et al*., 2005) proposed a decision support system based on an integrated knowledge base for acquisitions and mergers. It not only provides information concerning merger processes, major problems likely to occur in merger situations, and regulations practically or procedurally, but it also gives rational suggestions in compliance with the appropriate regulations. It can also suggest to the user how to deal with an uncertain growth rate and current evaluations.

The need effectively to integrate decision-making tasks with knowledge representation and inference procedures has meant that more and more recent research efforts have been angled towards the integration of decision support systems with knowledge-based techniques (Nikolopoulos and Assimakopoulos, 2003). A knowledge-based system (KBS) is a computer application that uses artificial intelligence techniques in problem-solving processes to support human decision-making, learning, and action (Akerkar and Sajja, 2010). However, KBS expands the range of tasks that can be automated under the supervision of human workers. Unlike conventional DSS which generally provide decision makers with information and tools to better analyze a problem, KBS also provides advice and recommendations (Dhaliwal and Benbasat, 1996).

In the simplest KBS, shown in Figure 2.6, there are two modules namely the knowledge base and inference engine. Knowledge base contains different rules which may be complex, and the facts may include sequence and structured entities. The inference engine will be responsible for generating the output or conclusion based on the type of inference approach used. In more complex systems, the inference engine itself may be a KBS containing meta-knowledge, i.e., knowledge of how to apply the domain knowledge.

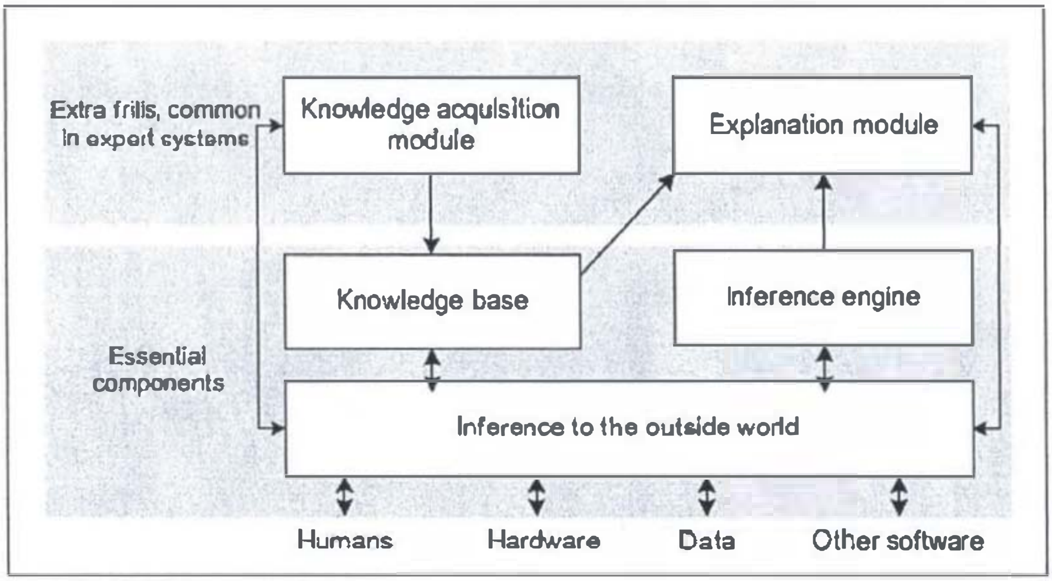

Figure 2.6 The main components of a knowledge- based system (Hopgood, 2000)

2.2.3 Business Intelligence (BI)

BI is a term that combines architectures, tools, databases, analytical tools, applications, and methodologies (Raisinghani, 2004), as shown in Figure 2.7. BI's major objective is to enable interactive access (sometimes in teal-time) to data, to enable manipulation of data, and to give business managers and analysts the ability to conduct appropriate analysis. By analyzing historical and current data, situations, and performances, decision makers get valuable insights that enable them to make more informed and better decisions (Zaman, 2005). The process of BI is based on the transformation of data to information, then to decisions, and finally to actions (Turban, Aronson *et al.*, 2006). The major benefit of BI to a company is the ability to provide accurate information when needed, including a real-time view of the corporate performance and its parts. Such information is a must for all types of decisions, for strategic planning, and even for survival.

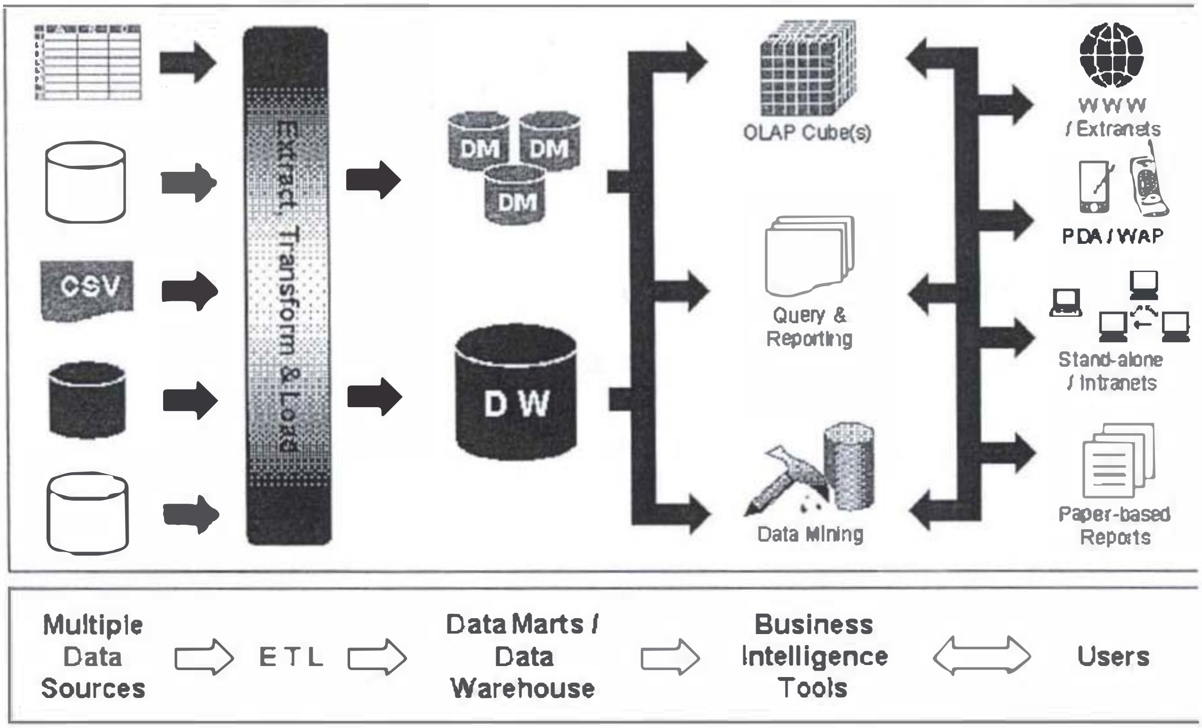

Figure 2.7 Business Intelligence Tools in an Enterprise

Since the accurate real-time information is the most valuable thing to a modern enterprise, it is essential to store, manage, and make use of it to facilitate knowledge discovery (KD) and decision support. Data Warehousing (DW) and On-Line Analytical Processing (OLAP) are two of the most significant new technologies in the business data processing arena (Datta and Thomas, 1999). Srinivasa (2001) proposed a business intelligent (BI) tool for helping logistics service providers to achieve strategy fonnulation. The BI tool comprises data warehousing, Online Analytical Processing (OLAP) and its related support system, which analyzes data collected from various sources and then converts it into actionable information. The common trait of business intelligence tools is that they contain certain data management software like data warehouse, OLAP and Multidimensional database management system (MDBMS) to store and explore the essential data in ways that are decision-oriented.

There are some differences between DSS and BI (Turban, Aronson *et al.*, 2006). Most DSS are constructed to directly support specific decision making, while BI systems, in general, are geared to provide accurate and timely information, and they support decision support indirectly. DSS are generally built to solve a specific problem and include their own databases, whereas BI applications focus on reporting and identifying problems by scanning data extracted from a data warehouse.

Recently there has been great interest in BI, explained by users' desires to gain competitive advantage. BI play an increasingly critical role in today's ever faster, ever more global, every more competitive business environment. To meet these needs, BI systems must evolve to be real-time, proactive and pervasive. It is reported that in the coming years, millions of people will use BI visual tools and analytics every day (Baum, 2006).

2.3 Related Studies on Existing Techniques for Intelligent Decision Support Systems

During the last two decades, a large number of emerging techniques mentioned in Table 1.1 have been developed to help the organization become more competitive in the global market. Every type of technique has certain capabilities and limitations. To assist managers in solving managerial or organizational problems faster and better, a DSS may integrate several of the tools and techniques as a hybrid (integrated) approach (Turban, Aronson *et al*., 2006). In addition to performing different tasks in the problem-solving process, these techniques can support each other. For example, an ES can enhance the modelling and data management of a DSS. A neural computing system can support the knowledge acquisition process in building an ES. ES and ANN play an increasingly important role in enhancing other decision support technologies by making them smarter. The components of such systems may also include OR, statistics, and a variety of other computer-based tools.

In recent years, intelligent systems, such as, ANN, GA and intelligent agents are considered as key techniques for managerial decision support in different industrial applications (Henrik, 1996). Using real-time data warehousing in conjunction with decision support systems is also an important way to support critical business processes.

In the following section, a wide range of techniques for intelligent DSS and their applications in industries will be discussed in order to address the future trend of intelligent DSS for Mobile Enterprise Management.

2.3.1 Data Warehouses (DW)

Decision makers are increasingly demanding access to concise, real-time information about current operations, trends, and changes. DW has emerged as one of the most powerful technologies for decision support during the last decade (Ramamurthy, Sen *et al.*, 2008). The primary objective of a data warehouse is to supply the information required for business decision-making, which stands in sharp contrast to the traditional database technology for capturing operational data (Dolk, 2000). It was originally envisioned as a separate architectural component that import a mass of raw data from a legacy operation systems and external sources, and integrate them in a manner consistent with the organization's needs (Chow, Choy *et al.*, 2005). According to (Inmon, 1992), a data warehouse is not just a large collection of integrated, subject-oriented databases; it has also taken on a much broader role.

DW architectures can be broadly classified into three types: Enterprise-wide DW (EDW), conformed data marts and Operational Data Store (ODS). EDW provides an enterprise view of corporate data with a single, central storage of data (Inmon, 1992). In the second architecture, the DW is a logical collection of individual data marts, which are built one at a time (Kimball, Ross *et al.*, 2009). A data mart is narrower in scope than an enterprise DW catering to the needs of a department or a functional group of users. An ODS consolidates data from multiple source systems and provides a near-real-time, integrated view of volatile, current data, which is used for short-term decisions involving mission-critical applications rather than for the medium- and long-term decisions associated with an EDW (Inmon, 1999).

Data warehousing traditionally focus on assisting manager in making strategic and tactical decisions by using historical data (Turban, Aronson *et al.*, 2006). Increased data volumes and accelerating update speeds are fundamentally changing the role of the data warehouse in modern business. Making fast and consistent decisions across the enterprise requires more than a traditional DW or data mart that is not business critical (Haisten, 2005). More data, coming in faster and requiring immediate conversion into decisions, means that organizations are confronting the need for real-time data warehousing (RDW) because yesterday's ODS will not support existing requirements (Langseth, 2006). E-business has become a major catalyst in the demand for RDW (Armstrong, 2000).

Real-time data warehousing (RDW) is the process of assembling data from OLTP systems as and when events happen and are moved at once into the data warehouse. (Gonzales, 2005) presented different definitions for RDW. According to (Basu, 2003), the most distinctive difference between a traditional data warehouse and an RDW is the shift in the data acquisition paradigm. RDW permits the instant updating of the data warehouse and the elimination of an ODS. At this point, tactical and strategic queries can be made against the RDW to use immediate as well as historical data. People throughout the organization who interact directly with customers and suppliers will be empowered with operational information-based decision making at their fingertips. (Turban, Aronson *et al.*, 2006) presents the case of Continental Airlines that demonstrates a scenario in which a real-time data warehouse supported decision making, through analyzing large amounts of data from various sources to provide rapid results to support critical processes.

Despite the benefits of an RDW, developing one can create its own set of problems that relate to architecture, data modelling, physical database design, storage and scalability, and maintainability. One critical issue in RDW is that not all data should be updated continuously. Depending on exactly when data are accessed, even down to the microsecond, different versions of the truth may be extracted and created, which might be confusing (Basu, 2003; Peterson, 2003; Terr, 2004).

2.3.2 On-line Analytical Processing (OLAP)

OLAP is the data manipulation software for decision-making applications by providing a service for accessing, analyzing and viewing a large amount of data with high flexibility and efficient performance. In the traditional approach, when a user needs to retrieve information across a multi-table, the tables used for finding the specific information must be clearly defined by using the Structured Query Language (SQL). SQL involves complex statements for retrieving results across a number of tables. When a user needs to get the results from more tables, the statement will be even more complex. OLAP can organize data into a hierarchy that represents levels of details on the data. According to Dayal and Chaudhuri (1997), the OLAP cube can be used to explore data by performing a slice and dice on the data, drilling down and rolling up the database. In spite of processing the data analysis, it can be operated in real-time mode, meaning that the analysis can be performed in a real time manner. OLAP is a kind of data mining technique that widely supports the knowledge discovery of Knowledge Management (KM) initiatives (Chen *et al.*, 2001; Tseng, 2004; Huang *et al.*, 2005; Tseng and Chou, 2006).

However, OLAP does not have the ability to select the best and most precise solution; it only allows users to understand the current situation at any point of time or other aspects through simple querying.

The data warehouse and OLAP are interconnected with each other for decision-making purposes, where the data warehouse is responsible for data storage and handling, and OLAP captures stored data to form valuable information for decision-making.

2.3.3 Intelligent Systems and Optimization Methods

In addition to the use of data and mathematical models, some managerial decisions are qualitative in nature and need judgemental knowledge that resides in human experts. Thus, it is necessary to incorporate such knowledge in developing DSS (Turban, Aronson *et al.*, 2006). A DSS that integrates knowledge from experts is called an intelligent DSS (IDSS) that can enhance the capabilities of decision support. The foundation for building IDSS is the techniques and tools that have been developed in the area of artificial intelligence (AI), such as rule-based Expert System (ES), Case-Based Reasoning (CBR), Artificial Neural Network (ANN) and Genetic Algorithms (GA). Since the introduction of the Internet and the Web, intelligent systems have become armed with a Web-based architecture and friendly user interface. Given the explosion of information on the Web, autonomous agents and recommendation systems will become very popular.

There are also a few optimization methods used in DSS to solve various practical problems, including GA, simulated annealing (SA), taboo search (TS), and Scatter Search (SS). These optimization methods have been termed meta-heuristics. The word "meta" comes from the fact that these heuristics work in an iterative master process that guides and modifies the operations of subordinate heuristics by combining intelligence, biological evolution, neural systems, and statistical mechanics. Although they do not guarantee optimal solutions to optimization problems, they are able to find near-optimal solutions efficiently by exploring and exploiting the search spaces using learning strategies.

2.3.3.1 Intelligent Agents and Multi-agents System (MAS)

An intelligent agent (IA) is a computer program that helps a user with routine computer tasks. It performs a specific task, based on predetermined rules and knowledge stored in a knowledge base. (Wooldridge, 2009) suggested the following features of intelligent agents:

- Reactivity. Intelligent agents are able to perceive their environment and respond in a timely fashion to changes that occur in it in order to satisfy their design objectives.
- Proactiveness. Intelligent agents are able to exhibit goal-directed behavior by taking the initiative in order to satisfy their design objectives.
- Social Ability. Intelligent agents are capable of interacting with other agents in order to satisfy their design objectives.
- Autonomy. Intelligent agents must have control over their own actions and be able to work and launch actions independently of the user or other actors.

Although more advanced agents are capable of doing multiple tasks, it is likely that many future agent systems will really be multi-agents, the collections of different agents, each handling a simple task (Wooldridge, 2009). A major value of intelligent agents is that they can handle many routine activities that need to be done quickly in order to save time by making decisions about what is relevant to the user. With these agents at work, a competent user's decision-making ability is enhanced with information rather than paralyzed by too much input. Agents are artificial intelligence's answer to a need created by computers (Nwana and Ndumu, 1999).

Information access and navigation are today's major applications of intelligent agents, but there are several other reasons this technology is expected to grow rapidly. For example, intelligent agents are playing an increasingly important role in e-commerce (Murch and Johnson, 1999; Liang and Huang, 2000; Lee, Liu *et al.*, 2002; Wooldridge, 2009). During the past decades, lots of work has been done in this field. To provide better value-added services to buyers and sellers, (Kong, Li *et al.*, 2004) developed an application of e-commerce in construction material procurement. They utilized mobile agents and web services to allow information sharing among e-commerce systems. Through an E-Union server, communication and information between e-commerce systems can be facilitated. Primarily, a client registered in an E-Union first sends a HTTP/HTML request, which stores material searching criteria, to the E-Union server. The application server in the E-Union server then explains the meaning of the request and passes it to the SQL server to find E-Union members that have the specified type of material. Next, it sends a

SOAP request to appropriate sites. Upon receiving the SOAP request, each appropriate site sends a SQLXML request to the database server and gets the response in a SOAP message to the application server. The E-Union application server receives all the responses from all other appropriate sites, integrates them, and then sends them back to the client in HTTP/HTML format. (Shakshuki, Kamel *et al.*, 2000) proposed a three-tier multi-agent framework for assisting different users in locating, retrieving, and integrating information from the World Wide Web. The system is comprised of a user agent, a broker agent, and a resource agent. The user agent allows the user to interact with the system and fulfill users' and other user agents' requests. It accepts user's topic of interest and the associated constraints. The broker agent acts as an intermediary between the user agent and the resource agent. Using KQML, the broker agent identifies and matches agents based on their goals as well as interests. The resource agent accepts a query structure similar to that of the user agent. After identifying the information resources and downloading the relevant documents, the resource agent parses the document and extracts the topics and their qualities. (Lee, Park *et al.*, 2001) designed a multi-agent system for Internet information retrieval services. In order to get effective and correct results, they re-designed an existing retrieval agent with effective distribution and multi-processing. The integrated system consists of a range of agents with different functions. Each agent includes a facilitator as well as some minor agents. The facilitator acts as a communicator among related agent groups and manages its own agent group.

Many other multi-agent techniques have been found in (Berry and Linoff, 1997; Liang and Huang, 2000; Bigus and Bigus, 2001).

The advantage of agents can be even greater when a wireless computing environment is involved. For example, (Wei, 2002) described the application of mobile commerce agents in WAP-based services. Several companies have developed agents that support different activities along the supply chain (Villa, 2002; Pardoe and Stone, 2004).

Multiple agents can collaborate to achieve certain goals. More important and unlike routine tasks that can be automated, decision making involves complex set of tasks that requires integration of supporting agents (Bui and Lee, 1999). These agents are called DSS agents that are active and goal-oriented and possess certain capabilities such as performing tasks and communicating with others. MAS are a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solver. MAS have emerged as a powerful technology for dealing with cooperation and decision-making processes in distributed information system applications.

Recent years have seen a number of publications related to the multi-agent-based model. An agent-based framework for building decision support systems has been developed by (Bui and Lee, 1999). (Symeonidis, Kehagias *et al.*, 2003) developed an Intelligent Policy Recommendation multi-Agent system (IPRA) that introduces adaptive intelligence as an add-on for ERP software customization. The system increases its intelligence by the use of agent technology and data mining techniques. In addition, a prototype Multi-Agent Enterprise Resource Planning (MAERP) system is proposed that utilizes the characteristics and capabilities of software agents to achieve enterprise wide integration (Lea, Gupta *et al.*, 2005). There is also an agent-based intelligent infrastructure of Contingency Management System (CMS) using knowledge source concept and agent technology as an agglutinating center of the system (Sheremetov, Contreras *et al.*, 2004). Furthermore, an agent-based multi-contract negotiation system is proposed for global manufacturing supply chain coordination (Jiao, You *et al.*, 2006). These efforts have been made to design various intelligent multi-agent information systems for their respective isolated purposes. Although currently developed systems demonstrated diverse intelligence, their performance may not be accurate or complete enough for processing certain tasks.

2.3.3.2 Case-Based Reasoning (CBR)

Case-based reasoning (CBR) is one of the AI technologies that are widely used in knowledge capturing and representation. The philosophy of CBR is to use previous similar cases to help solve, evaluate or interpret a current new problem (Zeleznikow, 1996). This process is similar to the mechanism used by human experts for the analysis of new situations. In fact, experience is an important element in human expertise, CBR is thought to be a more psychologically plausible model of the reasoning of an expert than a rule-based model (Watson, 2002). According to (Riesbeck and Schank, 1989), the use of this approach is justified by the fact that human thinking does not use logic but is basically a processing of he right information being retrieved at the right time. Generally, CBR has been formalized as a four-step process (Aamodt and Plaza, 1994):

- retrieval of the most similar cases;
- reuse of the cases to attempt to solve the problem;
- revision of the proposed solution if necessary; and
- retention of the new solution as part of a new case

The foundation of CBR is a repository of cases called a case base that contains a number of previous cases for decision making (Shiu and Pal, 2004). A case is a free data format with a set of attributes containing words, numbers and symbols to represent solutions and a state of affairs. These attributes are indexed to facilitate case retrieval. During case retrieval, the CBR engine is based on a set of attributes to locate similar cases to deal with similar problems. If the case were not indexed the retrieval time would increase, affecting case retrieval efficiency. As concluded by (Watson, 2002), case index setting is important for efficient case retrieval. Currently, two common case retrieval methods, namely the nearest neighbour method and the inductive indexing method (Wess, Althoff *et al.*, 1994), are widely adopted.

There are several advantages of using CBR:

- CBR allows agent to quickly propose new offers without deriving them from scratch.
- CBR alerts agents to avoid repeating past mistakes.
- CBR helps agents to analyze the important of features and issues of SCM and leads to better future deal.
- CBR is flexible in handling supply and demand uncertainty and changes in relationship and strategy

CBR makes learning much easier and the recommendation more sensible. In fact, there are already many successful applications of CBR for managerial problem solving (Khan and Hoffmann, 2003; Kwon and Kim, 2004; Kwon and Sadeh, 2004). However, CBR approach alone is not adequate to solve the problem in case the number of rules or cases is huge. Therefore, frame type CBR approach is recommended by Noh et al. (Noh, Lee *et al.*, 2000). An intelligent methodology for managing tacit knowledge residing is proposed in organization by applying both cognitive map (CM) and CBR as a tool for storing CM-driven tacit knowledge in the form of frame-typed cases, and retrieving appropriate tacit knowledge from the case base according to a new problem. This approach is a more systematic way to arrange the case, and the relationship between cases and the conditions in triggering the cases can be easily identified.

2.3.3.3 Artificial Neural Network (ANN)

In general, machine learning involves adaptive mechanisms that enable computers to learn from experience, learn by example and learn by analogy. Artificial neural network (ANN) is one of the most popular approaches of machine learning. In fact, ANN (shown in Figure 2.8) is characterized by arrays of highly-interconnected cells, often arranged in layered structures, where each cell, or neuron, is roughly similar to the next (Negnevitsky, 2001). Each neuron has an adjustable weight factor associated with it and connected to all other neurons in the adjacent layer through the weighted connections. The responsibility of it is to provide the desired changes of parameters based on what the network has been trained on. Intrinsically, a sufficient amount of data sample is a key factor in order to obtain accurate feedback from the trained network. It can learn the relationship between data sets by simply having sample data represented to their input and output layers (Kolanoski, 1996), the training of the network with input and output layers mapped to relevant realistic values with the purpose of developing the correlation between these two groups of data will not, in principle, contradict the basic principle of neural network.

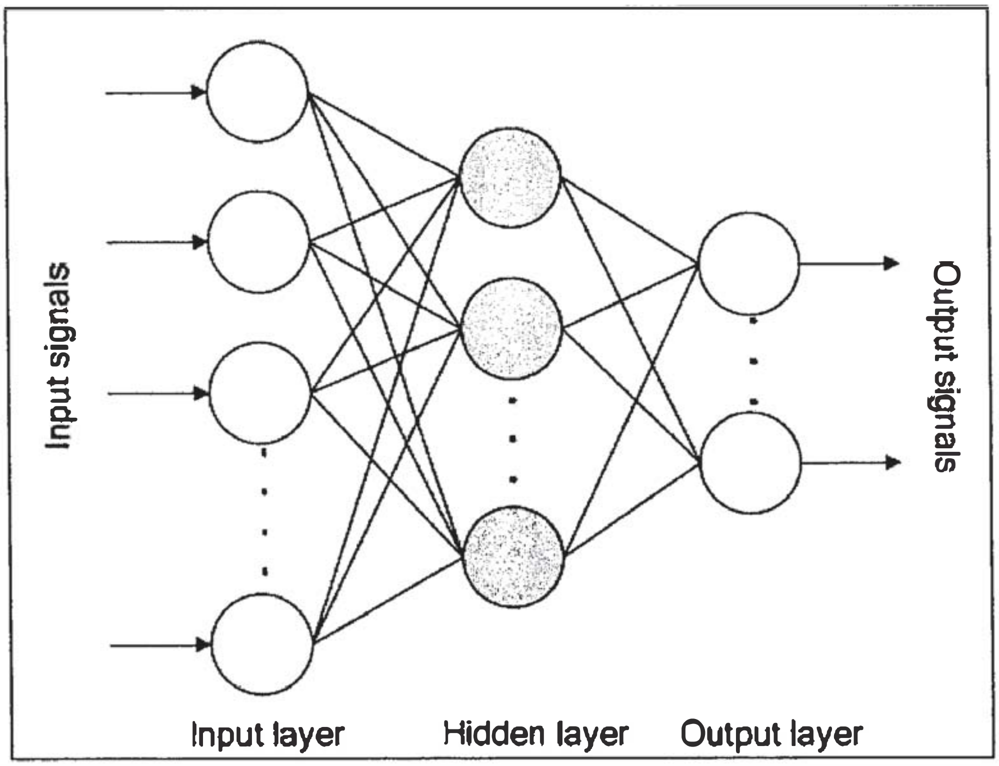

Figure 2.8 The architecture of typical ANN

Among neural networks, Multi-layer Perception (MLP) regression with an error back-propagation is commonly used in industries (Ho, Lau *et al.*, 2006). MLP have been applied to solve some difficult and diverse problems by training them in a supervised manner with an error back-propagation algorithm. This algorithm is based on the error-correction learning rule. The error back-propagation algorithm consists of two passes through the different layers of the network: the forward pass and the backward pass (Haykin, 1998).

Forward pass

$$V_j(n)= \sum_{i=0}^{m} W_{ji}(n)X_i(n) \qquad (2.1)$$

where *m* is the total number of inputs (excluding the bias) applied to neuron j, $W_{ji}(n)$ is the synaptic weight connecting neuron i to neuron j, and $X_i(n)$ is the input signal of neuron j.

Backward pass

$$W_{ji}(n) = \alpha \Delta W_{ji}(n-1) + \eta \delta_j(n) Y_i(n) \quad (2.2)$$

where α indicates momentum constant which controls the feedback loop acting around $\Delta W_{ji}(n)$, η indicates learning rate, $\delta_j(n)$ indicates local gradients that adjust the synaptic weights of the network. $Y_i(n)$ is the output signal of neuron i.

In recent years, ANNs have been reported to be very effective for solving many practical managerial problems. (Efendigil, Semih *et al.*, 2009) presents a comparative forecasting methodology regarding to uncertain customer demands in a multi-level supply chain (SC) structure via neural techniques. (Chang, Changchien *et al.*, 2006) proposes a method based on customer's on-line navigation behaviours by analyzing their navigation patterns through pre-trained artificial neural networks.

2.3.3.4 Genetic Algorithms (GA)

Genetic Algorithms (GA) was originally introduced by (Holland, 1992) as search algorithms and is a stochastic search algorithm based on the mechanics of genetics and natural selection. Basically, Genetic Algorithms are sets of computational procedures that conceptually follow steps inspired by the biological processes of evolution. The algorithm encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to generate new sample points in a search space. Many researchers are typically interested in genetic algorithms as optimization tools (Cus, Milfelner *et al.*, 2006). A flow chart for the basic GA is shown in Figure 2.9. In the basic algorithm, the following assumptions have been made:

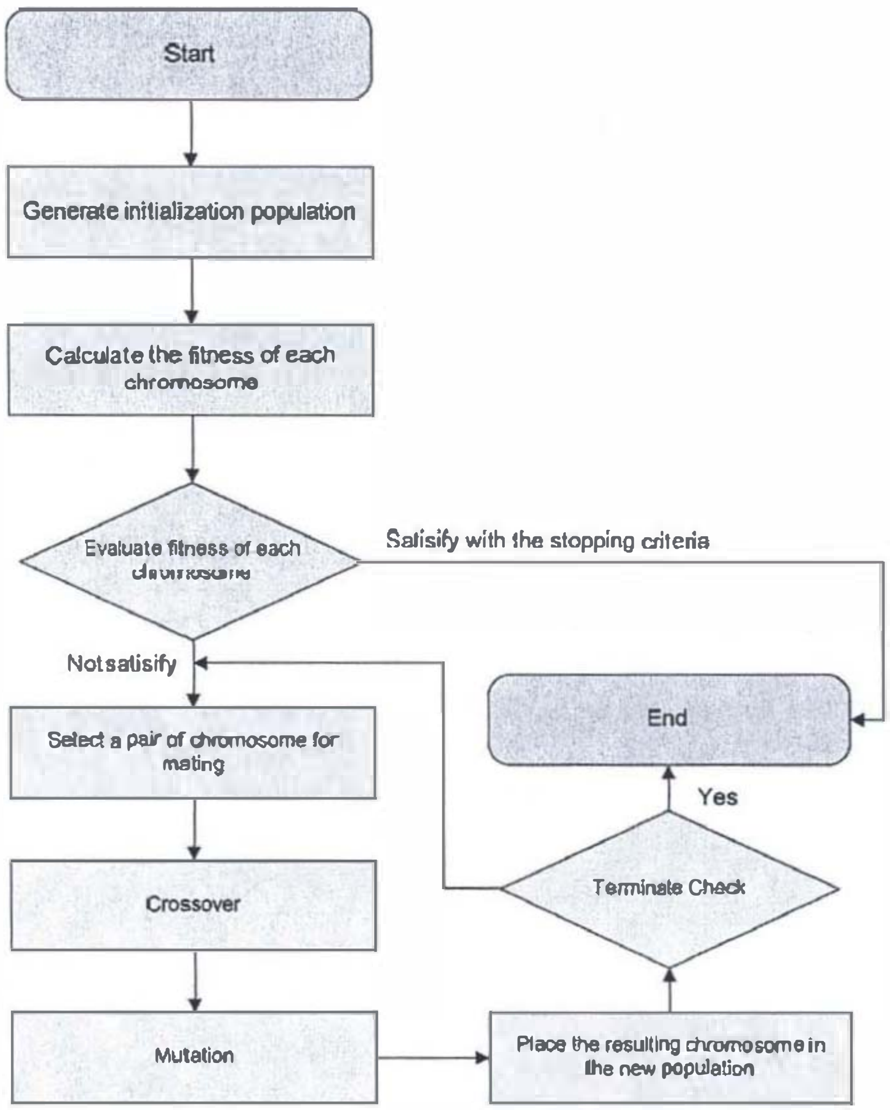

Figure 2.9 Flowchart of standard genetic algorithms

- The initial population is randomly generated.
- Individuals are evaluated according to their fitness function.
- Individuals are selected for reproduction on the basis of fitness; the fitter an individual, the more likely it is to be selected
- Reproduction of chromosomes, to produce the next generation, is achieved by "breeding" between pairs of chromosomes using the crossover operator and then applying a mutation operator to each of the offspring.

The GA has been used in a wide variety of business operation related tasks. (Kim and Ahn, 2008) proposed a clustering algorithm based on genetic algorithms (GAs) to effectively segment the online shopping market. An automated negotiator is developed that guides the negotiation process so as to maximize both parties' payoff (Choi, Liu et al., 2001). (Guan, Chan *et al.*, 2005) presented an approach in the detection of generic product attributes through feature analysis to provide an insight to the understanding of customers' generic preference. A genetic algorithm is used to find the suitable feature weight set, hence reducing the rate of misclassification.

2.3.3.5 Simulated Annealing (SA)

Simulated annealing (SA), first proposed by (Kirkpatrick, Gelatt *et al.*, 1983), is an optimization technique analogizing the thermodynamics process of annealing in physics. SA starts with an initial solution and repeatedly generates a new solution from the neighbourhood. If the new solution is better, it is accepted as the current solution. If it is worse, the new solution may be accepted and the acceptance depends on the acceptance function, the temperature parameter, and the difference in the objective values of the two solutions. Initially, the temperature parameter is large, and the new solution is accepted quite frequently. As the algorithm progresses, the temperature is slowly reduced, lowering the probability that the acceptance function will accept a worse solution. Figure 2.10 summarizes the general SA procedure (Pham and Karaboga, 1998).

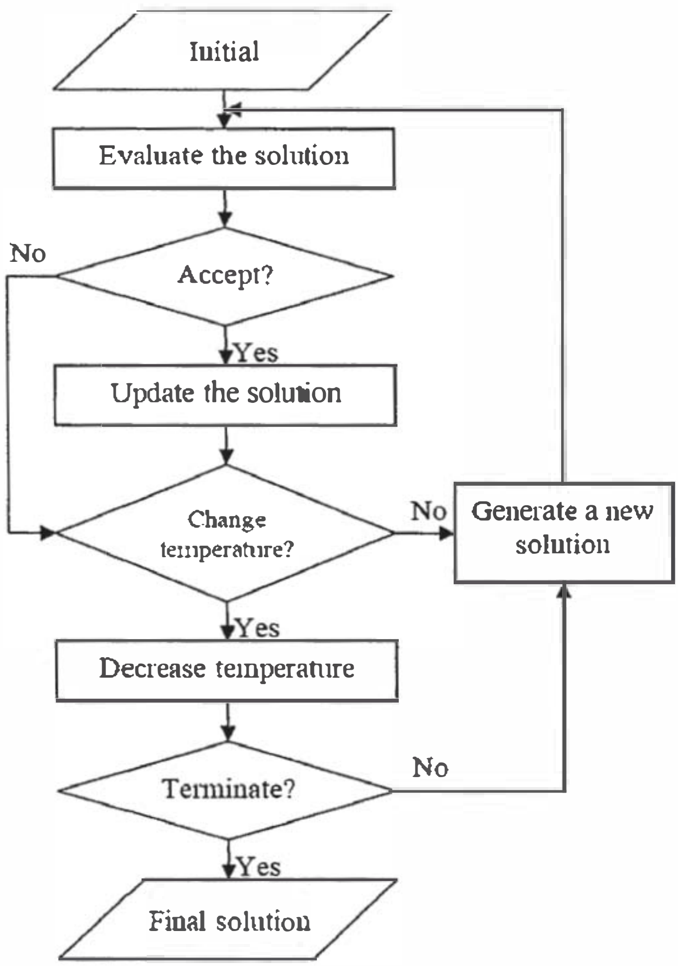

Figure 2.10 Flowchart of a standard simulated annealing method

2.3.3.6 Taboo Search (TS)

Taboo search (TS), primarily suggested by (Glover, 1995), is a strategy for solving combinatorial optimization problems by using especially designed memory structures to escape from the local optima. Like all other neighbourhood search techniques, TS starts with an initial solution and evaluates all its neighbourhood solutions. The best solution in the neighbourhood will be selected to replace the current solution even though it may not be better than the current one. In some occasions, the best neighbourhood solution may not be selected if the solution is in the taboo memory lists. There are two classes of memory lists: recency (short term) memory and frequency (long term) memory. Both memories are responsible for recording the history of the search and storing the forbidden moves (attributes). This mechanism attempts to prevent cycling behaviour and to force the search to new solution regions. If the selection is forbidden (taboo), the second best neighbourhood solution will be chosen as the candidate to update the current one. Also there is such a case that a taboo move may be accepted if certain criteria, called aspiration criteria, are met, such as the solution obtained by the application of the move being better than the best solution found so far. Then the newly updated solution is set as the primal for the next iteration. The search process continues until the stopping rule is satisfied. The flowchart of a standard TS method is illustrated in Figure 2.11 (Glover, 1995; Glover, 1998).

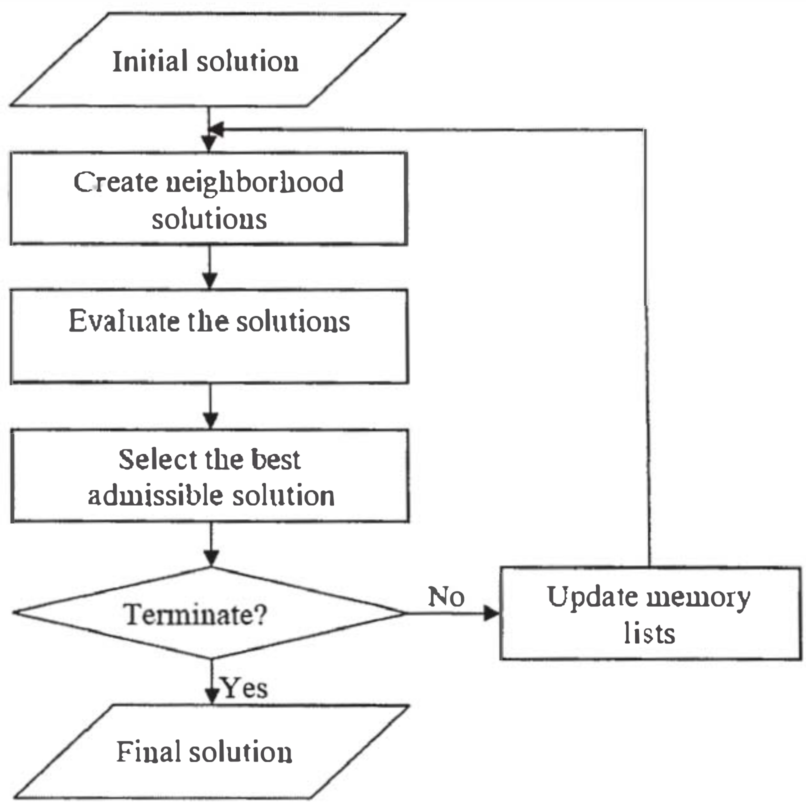

Figure 2.11 Flowchart of a standard taboo search method

2.3.3.7 Scatter Search (SS)

Scatter search was first introduced in (Glover, 1977) as a heuristic for integer programming, and it is quite an established meta-heuristics that has received renewed attention over the last decade. From the meta-heuristics classification, scatter search may be viewed as an evolutionary (or also called population-based) algorithm that constructs solutions by combining others and the goal of this methodology is to enable the implementation of solution procedures that can derive new solutions from combined elements (Glover, 1998). The scatter search methodology is very flexible, and each of its elements can be implemented in a variety of ways. Moreover, the algorithm incorporates procedures based on different strategies, such as diversification, local search, tabu search or path relinking (Haq, Saravanan *et al.*, 2007). Hence, Scatter search has recently been applied successfully to solve various complex problems (Jennings, 2000; Laguna and Marti, 2002).

2.3.4 Knowledge management System (KMS)

Competing in the globalized economy and markets require quick response to customer needs and problems. Knowledge is the major driving force for organizations' competitive capacities and wealth creation, and it is defined as a justified belief that increases an entity's capacity for effective action (Nonaka, Toyama *et al.*, 2000). (Liao, 2005) claimed that decision support is the objective of applying data mining to extract knowledge from a database for certain management issues. The need to effectively integrate decision-making tasks with knowledge representation and inference procedures has meant that more and more recent research efforts have been angled towards the integration of decision support systems with knowledge-based techniques (Nikolopoulos and Assimakopoulos, 2003).

Knowledge management (KM) is the systematic and organizationally specified process of creating, gathering, organizing, disseminating, and using knowledge so that employees can use it to become more effective and productive in their work (Artail, 2006). The structuring of knowledge enables effective and efficient problem solving, dynamic learning, strategic planning, and decision making. KM initiatives focus on identifying knowledge, explicating it in such a way that it can be shared in a formal manner, and leveraging its value through reuse. Knowledge management system (KMS) refers to a class of IT based information systems developed to support KM in organizations (Holsapple, 2003; Park and Kim, 2006; Sedighi, 2006; Zhang and Zhao, 2006).

Unlike conventional DSS which generally provide decision makers with information and tools to better analyze a problem, KMS further provides advice and recommendations. KMS plays an important role in supporting the intra- and inter-firm knowledge work and effectuating organization learning of the firms (Davenport and Prusak, 2000). The domain of KMS attracts numerous researches not only in academic prototype system level but also in industrial applications level (Lin and Tseng, 2005).

In recognizing the importance of KMS, leading world class organizations have been developing and utilizing their knowledge systems ranging from a single application to enterprise-wide application (Lee and Hong, 2002). A KM system of thinking has been developed, providing suggestions for what a general framework should include (Rubenstein-Montano, Liebowitz *et al.*, 2001). It is found in the literature that a number of KMS using different technologies have been developed (Davenport and Prusak, 2000).

The knowledge recommender is one kind of KMS for knowledge acquisition and sharing, which has become a promising and hot area in both academia and industries (Zhen, Huang *et al.*, 2010). Additionally, an Intelligent Policy Recommendation multi-Agent system (IPRA) is proposed that introduces adaptive intelligence as an add-on for ERP software customization, which takes advantage of knowledge gained through the use of data mining techniques, and incorporates it into the resulting company selling policy (Symeonidis, Kehagias *et al.*, 2003). As to the knowledge grid environment, a conceptual model of proactive knowledge recommendation was proposed in (Zhen and Jiang, 2008), and reactive knowledge query model for supporting innovation was proposed in (Zhen, Jiang *et al.*, 2009). However, most researches about the recommender systems mainly concerned with the recommendation algorithms, or recommenders for general situations in daily lives, seldom cover the recommendation scenarios for solving management problems in modem enterprise.

On the other hand, there are a number of mobile knowledge management systems for decision support. The use of context knowledge has been envisioned as an appropriate solution to deal with information overload matter: system responses can be summarized and customized depending on the situation and the preferences of the user, which results in presenting only the relevant information. Bobillo et al. (Bobillo, Delgado *et al.*, 2008) proposed a formal model for representing in ontologies relevance relations between context descriptions and domain-knowledge subsets. Yang (Yang and Wang, 2007) introduced the fractal summarization model for document summarization on handheld devices. Fractal summarization is developed based on the fractal theory. It generates a brief skeleton of summary at the first stage, and the details of the summary on different levels of the document are generated on demands of users. Such interactive summarization reduces the computation load in comparing with the generation of the entire summary in one batch by the traditional automatic summarization. In summary, most of current Information and Knowledge Based Systems could summarize and customize information for user's query; however, there is a lack of control function to automatically handle the user's instructions.

2.4 Secure Mobile Service Platforms

New avenues of telecommunication, such as high-speed wireless Wi-Fi networks, cellular phone networks and high-speed telecommunications service, coupled with entirely new hardware platforms, such as smart phones, PDA, and very powerful wireless laptop computers, are changing how people work, where they work, and what they do when they work.

Recently, a few companies and researchers are attempting to integrate the secure mobile service platform with enterprise information systems (EIS) to provide a new solution – the "mobile enterprise". From wikipedia, "Mobile enterprise is a collection of Online Interactive Business Applications made possible by Mobile Broadband. With the Mobile enterprise platform in place, entire businesses can be easily, quickly and economically moved onto the internet. Enterprise databases can be remotely accessed and updated from anywhere in the world, at anytime, with any device equipped with a Web Browser and by anyone with permission to access the service. Mobile Enterprise is real-time and fits the JIT business strategy."

Although the "mobile enterprise" concept has not been defined precisely by far, there are currently many secure mobile platforms available in the market to run business applications anytime and anywhere. The term secure mobile service platform refers to the protection of critical corporate resources - data, processes and network equipment - accessed by applications deployed on mobile devices and beyond the security perimeter of an enterprise.

AT&T labs proposed iMobile EE, which allows resource-limited mobile devices to communicate with each other and to securely access corporate contents and services (2003). Since the iMobile EE architecture is independent of specific access devices, wireless networks, or operating systems, it can very easily accommodate new emerging mobile devices and network solutions.

SPECSA is a scalable, optimized, policy-driven, extensible, and customizable security architecture developed for wireless enterprise applications (2004). It supports end-to-end client authentication, data integrity and confidentiality between wireless clients and enterprise servers. The security services provided by SPECSA are customized and controlled by an easily configurable security policy that specifies several security-related attributes, classifies network data based on sensitivity and content, and provides an abstraction for the communication and messaging between the client and the server.

W. Thomas (2004) et al. present a framework offering solutions for the development of secure mobile business applications that takes into account the need for strong security credentials. This framework consists of software and abstractions that allow for the separation of the core business logic from the security logic in applications. Security management instruments in the form of enforceable enterprise policies are defined which target the security and trust-related deployment and configuration of mobile devices and business applications.

Remote Access Virtual Private Networks (VPNs) are maturing at a time when secure network accessibility is becoming increasingly important to companies with multiple branches, telecommuters or a mobile workforce. IPsec can be used to support Remote Access VPNs by tunneling from an individual host to a security gateway. Up to now, IPsec support for roaming hosts is still limited. Client software must be installed on each IPsec host. Security parameters must be configured at each host, and each host must bear the additional processing imposed by IPsec. The hosts can be authenticated with certificates, which provides strong authentication but adds deployment complexity.

A Web-based monitor and management system architecture for enterprise virtual private network is proposed by R. S. Chen et al. (2005). This system architecture integrates both VPN devices and general network devices. Such feature can integrate the monitor and manage the VPN and Intranet at the same time. The result of this research provides enterprises with VPN architecture for monitoring and managing the enterprises, as well as to establish a prototype for the enterprise VPN monitor and management system.

Moreover, there are many IT companies, (i.e. Sybase, Mobile Enterprise System Limited, FierceMobileIT, PalmOne, HP) that provide commercial software packages to empower the mobile workforce with secure access to business applications on the go.

Overall, secure mobile service platforms are taking on greater importance in the enterprise, generally by accessing specialized Web servers that provide data and communication directly to the wireless device.

2.5 Data Acquisition Methods

There is a large amount of information in any modern enterprise, which can take various forms. Data is the raw material that is used to generate useful information and helpful knowledge (Lau, Wong *et al.*, 2003). Without up-to-date shop-floor data, it is impossible to make accurate decisions, no matter how advanced ERP systems and manufacturing equipment are (Huang, Zhang *et al.*, 2007).

During the last decade, a number of management information systems such as Supply Chain Management (SCM), Customer Relationship Management (CRM), Material Requirement Planning (MRP)/Manufacturing Resource Planning (MRPII), and ERP have been implemented successfully in different functional areas, each with its own database and data architecture. Besides the existing databases within a company, accurate and real-time raw data from various departments such as production and warehouse is also a vital input for decision support. In a manufacturing enterprise, real-time information on production orders, production quantities of each workstation and the whole production line, operative efficiency, etc., can be automatically collected from shop floors or assembly lines by using various types of data acquisition methods, including the manual recording method, barcode scanning, and the most updated radio frequency identification (RFID) technology (Guo, Wong *et al.*, 2009).

Therefore, the issues and mechanisms involved in the collection of the diversified real-time information from of a company can be divided into two types: One is direct information retrieval from the existing update databases that are linked with different departments; the other is automated data capture from real-time operations such as data entry by operators, process control computers and other monitoring devices (Lu and Sy, 2009). This section emphasizes on the review of widely adopted real-time data acquisition approaches such as barcode readers and RFID technology.

2.5.1 Barcode Readers

The barcode technology is to detect the optical signals reflected from barcode labels (Vrba, Macurek *et al.*, 2008). (Geoffrey Okogbaa, Huang *et al.*, 1992) presented the development of a data management system (DMS) linked to a distributed barcode based data acquisition network which facilitates the automation of the data acquisition process. In addition, a smart barcode detection method is proposed to utilize the specific graphic features of barcodes for positioning and recognition purposes (Youssef and Salem, 2007).

Furthermore, 2D-barcode and image recognition are more enablers for automated real-time data capture (Schuh, Gottschalk *et al.*, 2007). According to (Chen, Lu *et al.*, 2003), the main benefits of applying the 2D barcode system are that one piece of 2D barcode can contain more messages, and that the 2D barcode can continuously transfer the information to the next station while the network system being interrupted . The production status can be recorded in real-time using a barcode-reading system (1D and 2D), which is also a supplementary monitoring system (Chen, Lu *et al.*, 2003).

In addition to well-established barcode readers, RFID technology has matured in recent years, which will be discussed in the next section.

2.5.2 Radio Frequency Identification (RFID)

As the application of the RFID technology has become economically feasible, some RFID-based data capture systems have been developed to obtain real-time and accurate production data and their effectiveness has also been proved by various industrial applications and practices (Guo, Wong *et al.*, 2009).

The RFID system consists of three basic components: a tag, a reader, and back office data processing equipment. The tag contains unique identification infonnation about the objects or people to which it is attached; the reader emits and receives radio waves to read the infonnation stored in the tag, and then the data-processing equipment processes all the collected data (Wu *et al.*, 2006).

RFID tags are primarily classified as passive, semi-passive, or active, depending on the manner in which they derive operating power to run the digital logic on the chip and transmit the stored data to the reader. With an independent power supply, active RFID tags allow greater communication range, higher carrier frequency and data transmissions rates, better noise immunity, and larger data storage capacity as compared with passive tags. However, the trade-off is a finite lifetime (optimally, 5 or more years), and greater size and higher cost compared with passive tags (Sarma and Engal, 2003). Different frequencies and types of RFID tags are adopted to identify or track the object and people. The tag reader, the other component of the RFID system, uses the method of inductive coupling to communicate with tags. The coiled antennas of readers create magnetic field with the tag's antennas that transmit energy and send back waves to the reader. The software aggregates the data read from the readers, cleanses it for errors or duplication, and forwards it to the appropriate enterprise systems or database.

The decrease in the cost of RFID technology (tags and readers) has motivated worldwide sporadic piloting efforts across different product sectors (Huang, Zhang *et al.*, 2007). Chappell et al. (2003) provides general overview on how RFID technology can be applied in manufacturing. RFID technology, whether deployed to individual part items (Kiritsis, 2004; McFarlane et al., 2003; Zaharudin, 2002) or their containers, is able to capture real-time field data from the workshops. (Huang, Zhang *et al.*, 2007) indicated that the emerging advanced wireless manufacturing technology has relied substantially on wireless devices such as RFID and wireless information networks for the collection and synchronization of the real-time field data from manufacturing workshops. (Guo, Wong *et al.*, 2009) proposed an RFID-based data capture system (as shown in Figure 2.2) that collects all real-time job processing records and production data from the flexible assembly lines (FAL). On the basis of the real-time production data stored in a database server, the decision support model generates effective solutions to production control on the FAL. Gupta (2000) addressed RFID is part of range of technologies used for automated data collection to augment enterprise resource planning (ERP) system activities. Chow *et al.*, (2006a) presented a RFID based resource management system using different types of RFID tag to collect dynamic and static logistics data. They used a hybrid AI system to diagnose these RFID data to formulate a real-time based tangible resource usage packages for use in warehouse operations ordering. RFID is being increasingly applied in inventory management (Smaros and Holmstrom, 2000) and SCM (Ngai *et al.*, 2005). Moreover, this emerging technology has also been further extended to the knowledge-based system or other systems for knowledge creation or decision-making support.

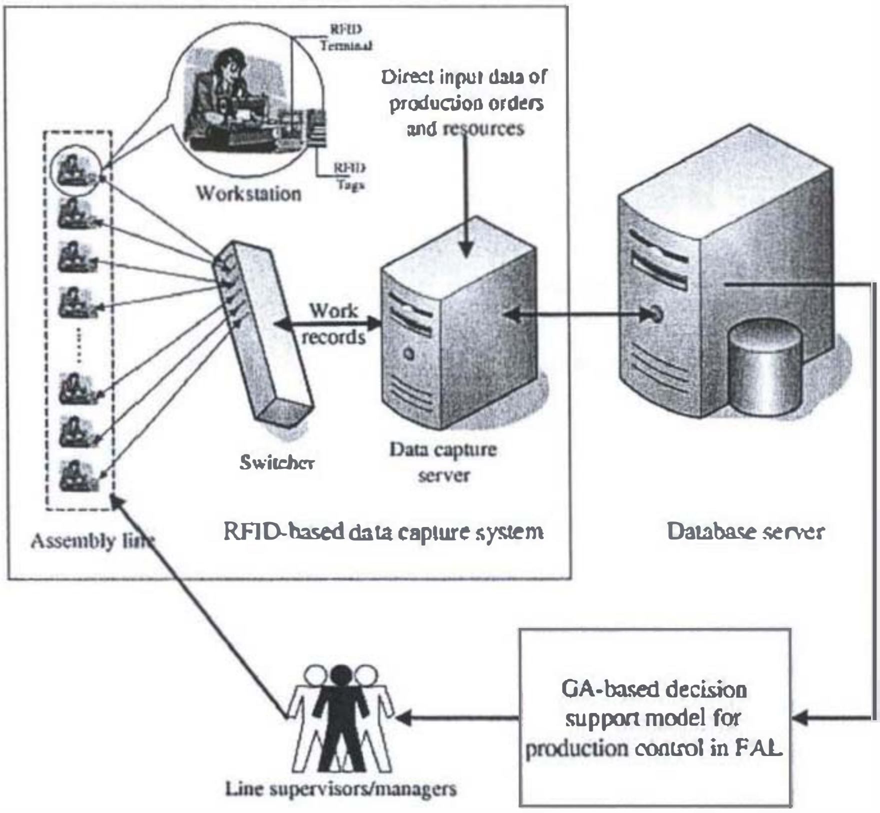

Figure 2.12 Architecture of intelligent production control decision support system for flexible assembly lines (Guo, Wong *et al.*, 2009)

A review of the publications about RFID technology illustrates its capability of automatic identification and data collection. However, the software products designed for the integration of the RFID data with enterprise wide applications in real-time decision-making activities are few and new in the market. As Beaulah *et al* (1998) illustrated, the performance of real-time signal processing and interpretation are directly affecting the usefulness of the real-time Knowledge Based System (KBS). RFID technology has the potential to take over the role of real-time based sensors to monitor and reflect the real-world situation of logistics processes and activities.

Recent efforts in business process management (BPM) have led to inputs directly from physical devices for analysis via DSS. For example, RFID chips can record data from sensors in railcars or in process products in a factory. Data from these sensors (e.g. recording an item's status) can be downloaded at key locations and immediately transmitted to a database or data warehouse, where they can be analyzed and decisions can be made concerning the status of the items being monitored (Huang, Zhang *et al.*, 2007).

2.6 Summary

During the recent years, many efforts have been made to develop various important specialized enterprise management information systems. At the same time, personal devices (PDAs, cell phones, tablet computers, laptop) incorporation with cellular wireless data networks has been a more and more matured technology, which can be integrated with management systems to develop a brand-new concept - "Mobile Enterprise". It is especially important for senior managers on the move who can be more effective and efficient if they can obtain the real-time and comprehensive status of their enterprises (such as inventory, capital, production and otherwise), and then respond with information updates, collaboration efforts, and decisions.

Actually, because there is no consensus on exactly what a mobile enterprise is, there is obviously no agreement on the standard characteristics and capabilities of ME. In terms of academic research, several fields related to the design of a mobile enterprise are reviewed. A number of emerging technologies directly and indirectly influence decision support systems. Data access, communication, and collaboration are critical in making DSS technologies work. A review of decision support and business intelligence systems revealed that the characteristics of current DSS only provided actionable information.

In summary, there are a number of approaches and systems, which have been designed and implemented to improve enterprise managerial decision-making. However, currently developed systems have their own limitations. They are not complete solutions for mobilizing an enterprise because their paradigms were originated from different backgrounds and were designed for their respective purposes. In our investigation, some of multi-agent systems lack control function, or can not be updated in real-time though they can perform certain tasks intelligently. Thus, these systems can only be important references for part of an innovative intelligent object paradigm for monitoring and control of mobile enterprises to cater for the future trend in wireless management information system especially for SMEs.

Based on the reviews, the limitations and problems in current DSSs are summarized to form the starting point of the study. In later chapters, a MMAP will be proposed to create a robust, adaptive, and cost-effective approach, and intend to be applied to various industries.

Chapter 3 The Mobile-enterprise Multi-Agent Paradigm (MMAP)

3.1 Introduction

This chapter focuses on the development of a Mobile-enterprise Multi-Agent Paradigm (MMAP) to provide a systematic and efficient approach to facilitate remote monitoring and control of modern manufacturing enterprises. Based on the distributed manufacturing point of view, various information systems are implemented in different departments to support different operations. The MMAP is proposed to coordinate different functional areas for the purpose of offering optimal and real-time result or recommendation for enterprise managers on the move through their mobile devices. On the other hand, it provides the function to automatically distribute the instructions given by the managers and execute it in an instant manner. In order to assemble all of the above functions, our approach is to distribute the functions to distinct agents in order to increase the flexibility and the adaptability of the systems. In this way, the multi-agents system (MAS) design for processing information is the core of our system. In the design of the architecture of MMAP, artificial intelligence (AI) plays an instrumental role in improving the decision making quality and accuracy by deriving useful knowledge from information processing.

In the following sections, the architectural structure of MMAP is described with the detailed explanation of two main modules.

3.2 Architecture of MMAP

The MMAP is designed to retrieve the distributed operation information from different functional departments within the enterprise and automatically formulate the information into the appropriate answer to suggest the optimized solutions which have significant influence on decision-making especially during critical business negotiations. It also allows managers to issue order instructions to the relevant part of the organization. The generic architecture of the MMAP is illustrated in Figure 3.1.

The paradigm consists of two modules, namely, Real-Time Data Warehouse (RTDW), and Query and Control Multi-agent Module (QCMM). RTDW is responsible for manipulating various real-time data within the enterprise using technologies of data warehouse and OLAP. QCMM is the core module of MMAP, the framework of which is developed by multi-agent technology. The module comprises three kinds of agents to provide optimized decision support for query and control request from end users. Individual query or enjoin agent is identified by the query or command type, while each data collection agent represents a functional department. The output of the system is displayed to the system user through a secure wireless platform, which is able to establish a secure, fast and smooth connection that allows applications on mobile devices to communicate with other electronic terminals within the Intranet through any number of firewalls and proxies.

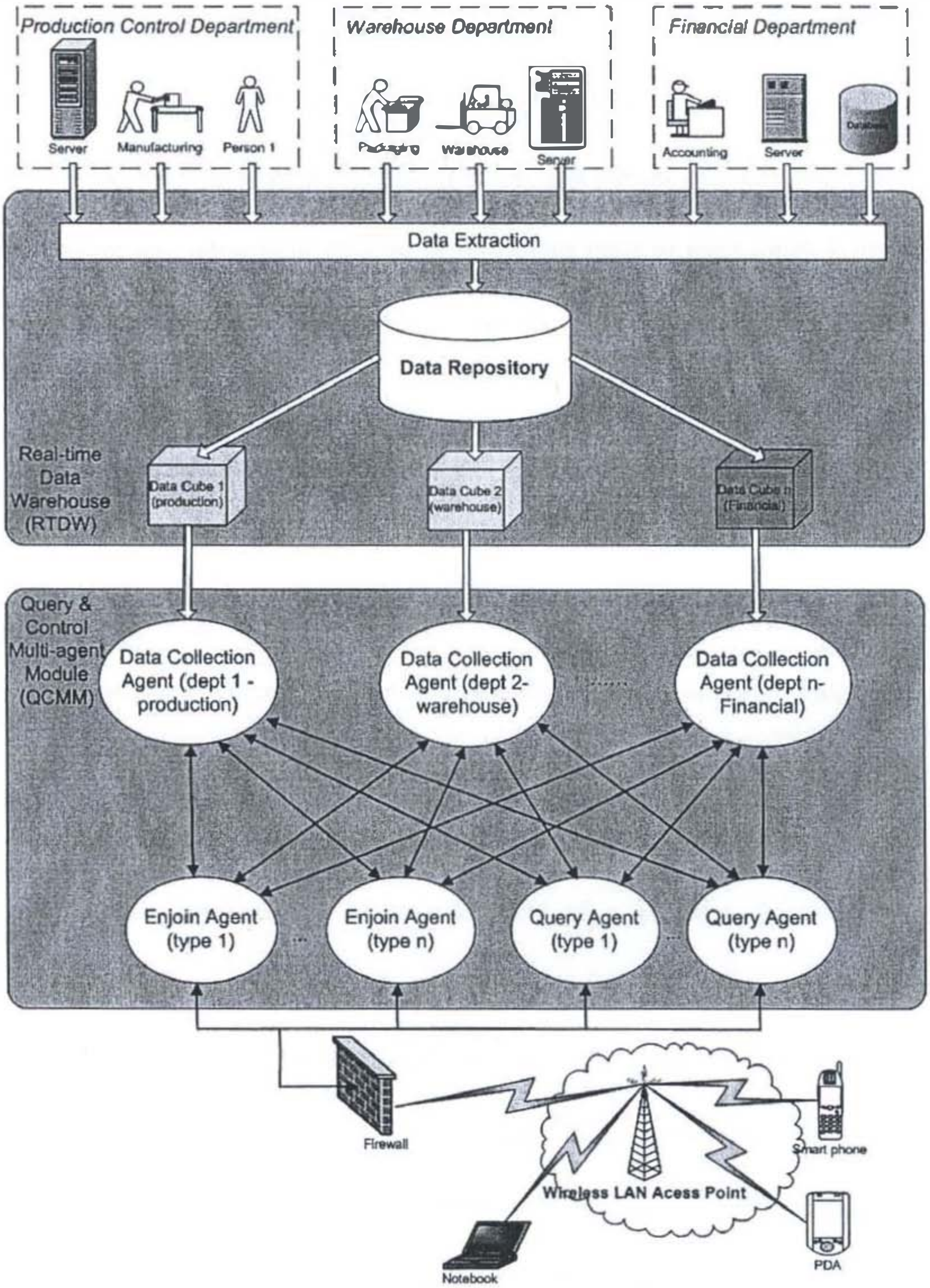

Figure 3.1 Generic Architecture of MMAP

3.2.1 The Real-Time Data Warehouse (RTDW)

As data warehouse and business intelligence applications have matured for strategic planning and decision-making, it has become apparent that the information and analyses they provide are vital to tactical day-to-day decision-making. Consequently, there is a trend towards integrating decision processing into the overall business process. The advent of e-business is also propelling this integration because organizations need to react much faster to changing business conditions in the e-business world (Inmon, Terdeman *et al.*, 2001).

New information is being generated continuously by operational sources such as order processing, inventory control, and customer service systems (Bruckner and Tjoa, 2002). The decision making process in traditional data warehouse environments is often delayed because data cannot be propagated from the source system to the data warehouse in time (Inmon, 1992). RTDW aims at decreasing the time it takes to make business decisions and tries to attain zero latency between the cause and effect of a business decision. It is responsible for collecting and manipulating different types of real-time information from various functional areas to support the functionality of the core module of QCMM. Therefore, it supports an immediate discovery of abnormal data conditions in a manner not supported by an OLTP system. The data may be collected in several ways: (a) legacy databases that are linked directly with different departments, (b) enterprise application systems, (c) the data entry by staff through Intranet environment, and (d) data capture tools such as RFID and Barcode located in the shop floor and warehouse. Business requirements may appear to be different across the various industries, but the underlying information requirements are similar – integrated, current, detailed, and immediately accessible. This requires efficient ETL (extraction, transformation, loading) techniques enabling continuous data integration, which enables real-time capturing and loading from different operational sources. In order to support efficient analysis and mining of such diverse and distributed information, the acquired real-time operation data is divided into subsets, summarized, categorized and stored in a central data repository for further action.

3.2.1.1 ETL Environment for RTDW

A continuous ETL environment for RTDW is deployed to facilitate better and faster decision making, resulting in streamlined operations, increased management performance in the business processes, improved customer relationships and enhanced e-business capabilities. The architecture is to accelerate the ETL process by moving data between the different layers. It is achieved by 1) extracting the data with high-speed J2EE connectors, 2) immediately parsing and converting the source data into the XML format, and 3) converting and assembling the data for the target format. In this way, each layer of the ETL environment can process the source data and forward it to other layers for further processing. Figure 3.2 shows the levels of a typical ETL process.

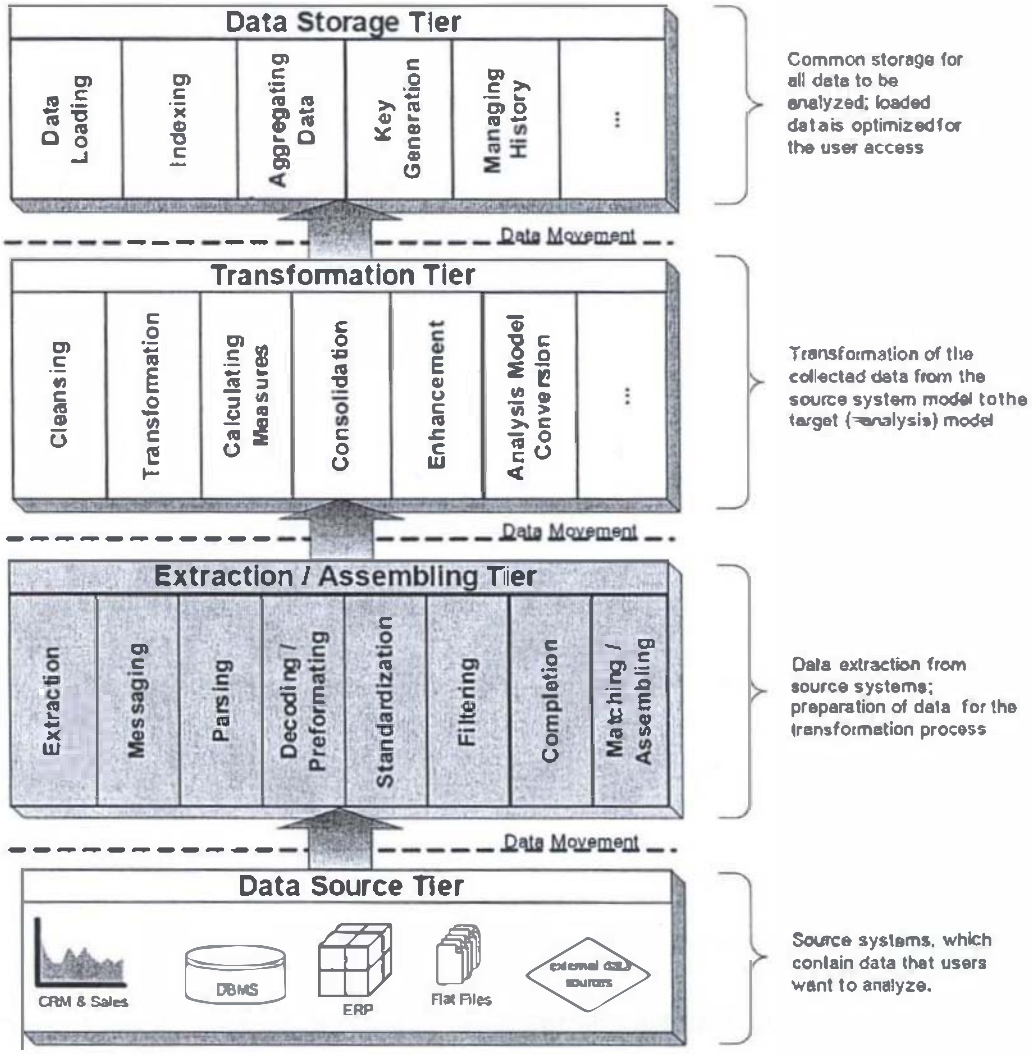

Figure 3.2 ETL processing layers of a real-time data warehouse

The processing steps for each layer may be executed in a distributed environment at different locations. The data propagation from the source system and the data movement within the ETL environment are managed by containers. The containers are used to ensure that the ETL components get instantiated for the processing and that the data is passed from one component to the next. One of the most critical ETL processing tasks is an efficient data extraction. An ETL environment of a RTDW must be able to establish high-performance connections to the source systems and optimize these connections. For instance, connections to databases or enterprise systems must be automatically pooled to optimize the resources and throughput times. In our approach, containers manage the connection pools to provide an efficient access to the data sources.

A J2EE (Java 2 Platform, Enterprise Edition) architecture is presented for the ETL environment of RTDW. For RTDW, a robust, scalable and high-performance data staging environment, which is able to handle a large number of real-time data propagations from the source systems is essential. J2EE is a Java platform designed for the mainframe-scale computing typical of large enterprises. Sun Microsystems (together with industry partners such as IBM) designed J2EE to simplify application development in a thin client tiered environment and to decrease the need for programming and programmer training by creating standardized, reusable modular components and by enabling the tier to handle many aspects of programming automatically (Singh, Stearns *et al.*, 2002). J2EE environments have a multi-tiered architecture, which provides natural access points for integration with existing and future systems and for the deployment of new sources systems and interfaces as needs arise. In J2EE environments, the container takes responsibility for system-level services (such as threading, resource management, transactions, security, persistence, and so on). This arrangement leaves the component developer with the simplified task of developing business functionality. It also allows the implementation details of the system services to be reconfigured without changing the component code, making components useful in a wide range of contexts. In our approach, we extend this concept by adding new container services. A container service is responsible for the monitoring of the data extracts and ensures that resources, workload and time-constraints are optimized. ETL developers are able to specify data propagation parameters (e.g. schedule and time constraints) in a deployment descriptor and the container will try to optimize these settings.

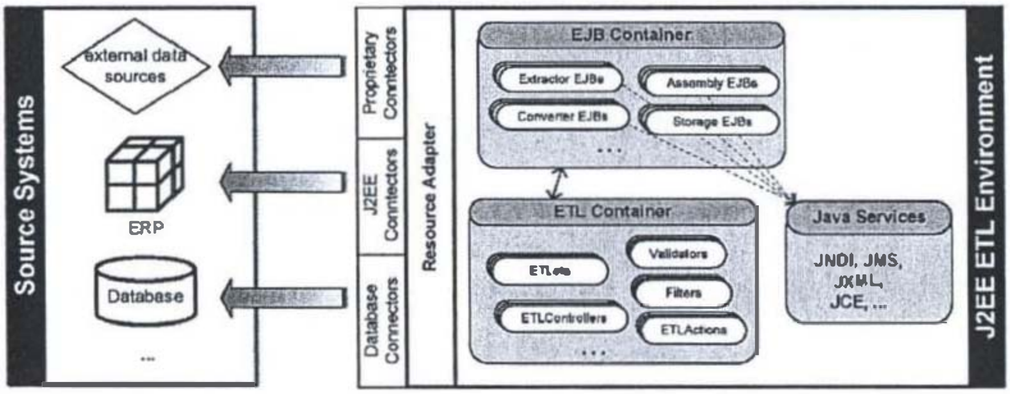

Figure 3.3 J2EE ETL environment

Figure 3.3 shows a J2EE ETL environment with resource adapters for the data extraction, which are available for the data propagation. Source systems for the ETL environment may require different types of access connectors. J2EE environments include a standard API for high-performance connectors. Many vendors of ERP or CRM systems (e.g. SAP, PeopleSoft) offer a J2EE connector interface for their systems. The J2EE platform includes synchronous and asynchronous communication mechanisms via J2EE connectors. Furthermore, the J2EE platform includes a standard API for accessing databases (JDBC) and for messaging (JMS) which enables the ETL developers to access queue-based source systems, which can propagate data via a messaging system. ETLets and Enterprise JavaBeans (EJB) components are configurable Java components which perform the ETL processing tasks, such as data extraction, validation, transformation, or data assembly. They can be deployed on any Java application server and can be reused for several ETL processes. An ETL container is a part of a Java application server that provides services for the execution and monitoring of ETL tasks. It manages the lifecycle of ETLets and provides default implementations for some of the interfaces shown in Figure 3.4. Like other Java-based components, ETLets are platform independent Java classes that are compiled to platform neutral bytecode that can be loaded dynamically into and run by a Java application server. ETLets interact with schedulers or ETL clients via a predefined interface of the ETL container. The ETL container supports filters and validators, which allow on-the-fly transformations and validations of data being processed. ETL filters can globally modify or adapt the data being processed. ETL validators are used to ensure data consistency and can be used to check business rules. Filters and validators are pluggable Java components that can be specified in the deployment descriptor for the ETL container. ETL controllers are Java classes that act as intermediaries between ETLets and the underlying ETL processing model. ETL controllers coordinate the ETL tasks by processing events with ETL actions, which use the EJB components (ExtractionBean, ConversionBean, AssemblyBean etc.) to perform the processing.

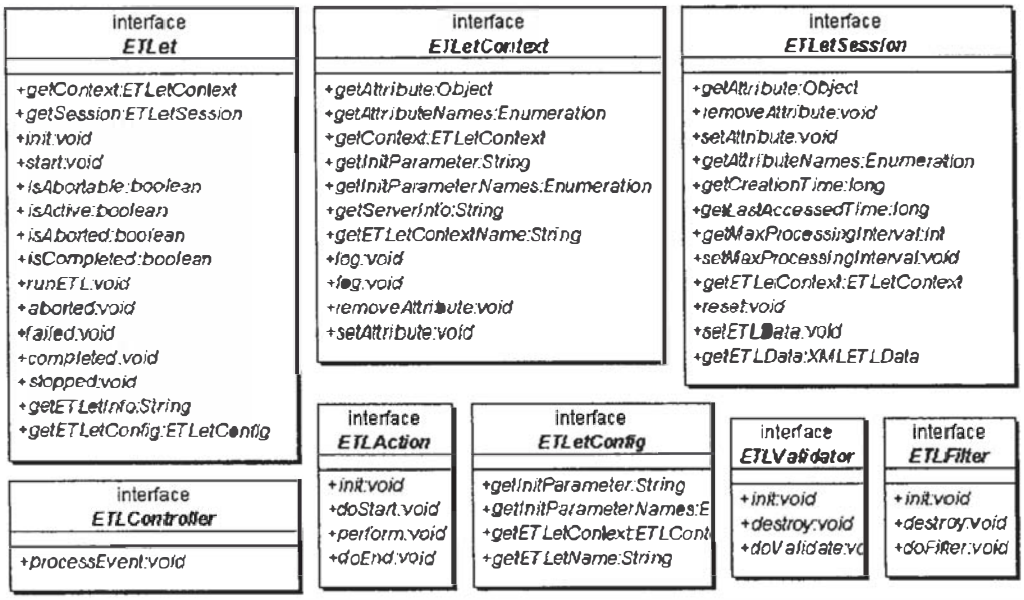

Figure 3.4 Java interface supported by the ETL container

Figure 3.4 shows the Java interfaces, which are supported by the ETL container. ETLets of the same type share an ETLet context, which is accessible to all instances of an ETLet class and can be used to find out initialization parameters or to share attributes among all ETLet instances. For instance, the ETLet context can be used for implementing caching strategies. The ETLetSession interface is used by ETLets to maintain the business data for one run of the ETL process. The method getETLData() makes the extracted data accessible to all components used for the ETL process. The ETLetConfig interface provides information about the initialization parameters for the ETLet. The ETLValidator and ETLFilter interfaces must be implemented by validator and filter components, which can be used in several ETL processes by specifying them in the deployment descriptor. ETL developers can use the

ETLController and ETLAction interfaces for implementing complex control logic for the ETL process. For simple ETL processes developers can simply implement the runETL() method of the ETLet. For more complex scenarios, developers can implement ETLActions to 1) encapsulate processing steps, 2) instantiate EJB components, and 3) to invoke methods from these EJB components. The ETLActions are invoked by an ETLController, which contains the centrally managed control logic.

The EJB container of the J2EE environment provides portable, scalable, available, and high-performance access to data and ETL processing. The Java application server manages the efficient access to instances of the EJB components regardless of whether the components are used locally or remotely.

3.2.1.2 Data Cubes Storage with OLAP

According to literature, OLAP can organize data into a hierarchy that represents levels of details of the data. In our approach, Multidimensional Database Management System (MDBMS) is deployed to support the OLAP operation, which is shown in Figure 3.5. The data that are stored in the relational data warehouse is first extracted by MDBMS and then transferred into the multidimensional data structure, which is the OLAP data cube. The cube is able to store aggregated data for providing a multidimensional view of data with high flexibility and good performance. It can be used to explore data by performing a slice and dice on the data, drilling down and rolling up the database. In spite of processing the data analysis, it can be operated in real-time mode, meaning that the analysis can be performed in a real time manner.

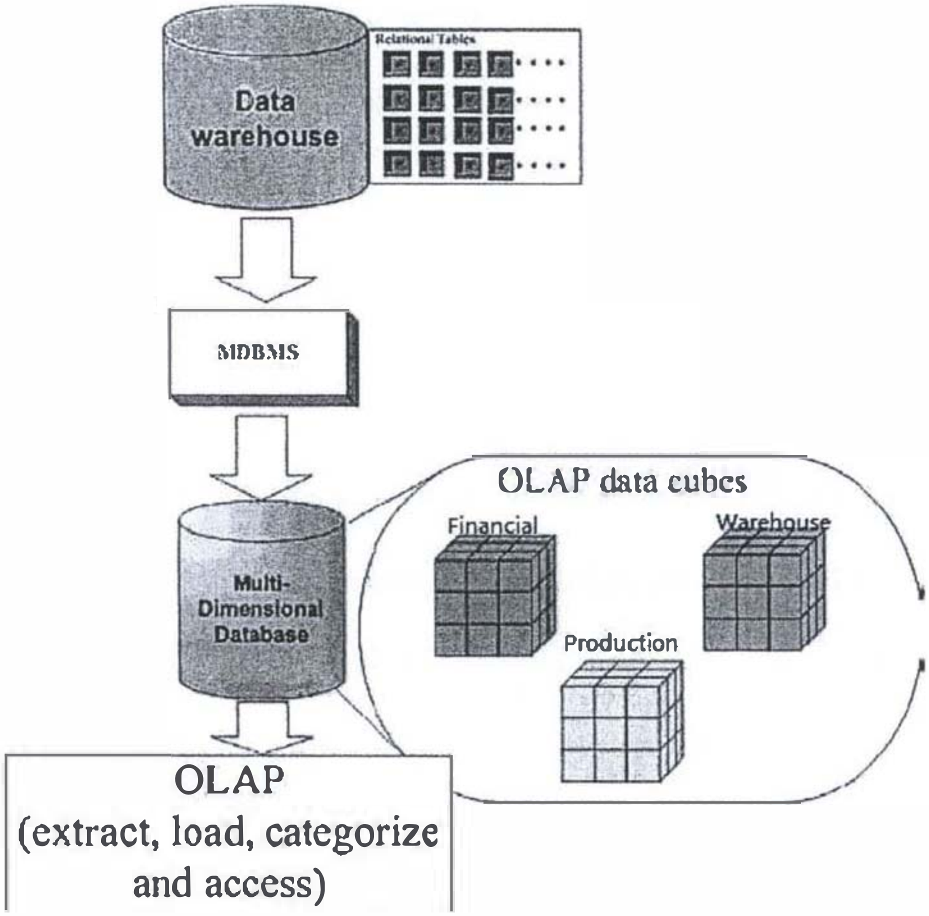

Figure 3.5 The data manipulation process of OLAP

The OLAP data model consists of descriptive data (dimensions) and quantitative values (measures) which build up the OLAP data cube. In the OLAP data cube, the data are built up using two elements: a fact table and a dimension table. In the fact table, the measures are defined for data analysis. In the dimension table, the different dimension levels are defined and used on different views of the OLAP data cube.

Since the data/infonnation in an organization is probably stored in an environment within the total control of the respective functional area for easy trading and update, it is appropriate that information be stored in individual OLAP data cube and be managed by their respective agents. The infonnation generated from OLAP is then passed to data collection agents in QCMM for further analysis.

3.2.2 The Query and Control Multi-agent Module (QCMM)

QCMM is the core module of MMAP that is responsible for providing optimized decision support for query and control request from end users. The framework of QCMM is developed by multi-agent technology. Every agent within the multi-agent technology has its particular mission(s). In order to achieve the mission(s), an agent is equipped with certain behaviour to perforn a specific activity autonomously. Each activity is carried out according to a set of predefined rules and procedures that represent the agent's action. In addition, in order to perform an activity, particular information is required. For example, job_id and process_id must be provided so that the agents recognize the process being executed. This scenario of multi-agent technology is expressed by the standard agent algorithms:

Agent algorithm:

Agent<agent_name>

Mission <mission_statement>

Activity$_1$<description>

Dataset<data>

Action<function>

... ...

Activity$_n$<description>

Dataset<data>

Action<function>

As each agent is not isolated but interacts with other agents to achieve the goals, communication is an essential capability that an agent must possess. The agents communicate with each other through exchanging messages with the function send_msg() which is defined as:

send_msg(String sender, String receiver, String request_action, Vector dataset) where:

sender: specifies the agent who sends the message **receiver:** specifies the agent who receives the message **request_action:** specifies the action to be performed by the receiver who handles the request **dataset(attribute(s)):** carries a set of data to enable the receiving agent to recognize the identity of the process ; also provides other important information such as the result of a particular activity

There are three kinds of agents in QCMM. In MMAP, when an enquiry/command comes in, the information needed would normally come from several distributed data cubes and hence a separate query agent or enjoin agent is required to interpret the enquiry/command and identify from which data source the respective information would come from and then issue the requests for information to the appropriate data collection agents accordingly. Since the data collection agents have intimate knowledge of the data storage that it searches, they will have first-hand knowledge of whether the data is available or not. In the following sections, the mechanisms of three agents DCA, QA and EA are illustrated.

3.2.2.1 The Data Collection Agent (DCA)

The objective of the DCA is to search its associated data storage within the functional area and obtain the information requested by query or enjoin agent. It possesses specific domain knowledge needed to carry out its tasks. The "intelligence" in the data collection agents identifies invalid data and missing values so that the data is complete and applicable when being returned to the query or enjoin agent. The architecture of the DCA is as shown in Figure 3.6.

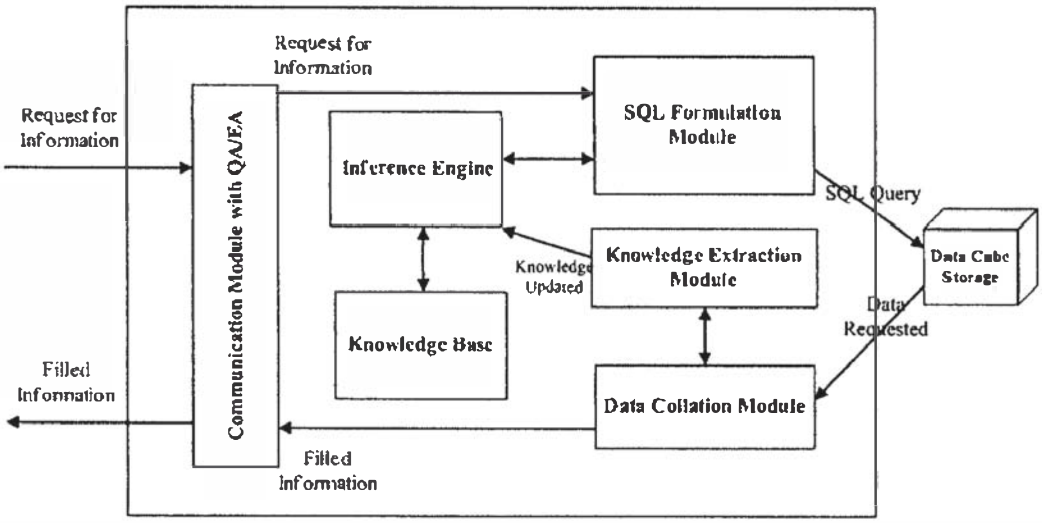

Figure 3.6 Architecture of Data Collection Agent

The responsibility of a DCA is to:

- receive request for information form by QA/EA
- form SQL query for relevant information from its associated Data cube storage using its own knowledge
- collate retrieved information
- update knowledge base by data mining

The DCA in MMAP accomplishes one or more of the following functions:

1. Description—Description provides a concise and succinct summarization of a collection of data and distinguishes it from others. Data description should cover not only its summary properties, such as count, sum, and average, but also its properties on data dispersion, such as variance, quartiles, etc.

2. Analysis—Analysis is to examine the large set of data to find certain regularities and interesting characteristics, including search for similar sequences or subsequences, and the sequential patterns, periodicities, trends and deviations.

3. Association—Association is the discovery of relationships or correlations among a set of data items. They are often expressed in rule forms showing attribute-value conditions that occur frequently together in a given set of data. An association rule in the form of $X \Rightarrow Y$ is interpreted as 'database tuples that satisfy X are likely to satisfy Y'.

4. Clustering-—Clustering is to identify attribute clusters embedded in the data, where a cluster is a collection of data objects that are 'similar' to one another. Similarity can be expressed by distance or some other functions specified by users or experts. A good clustering method must ensure that the inter-cluster similarity is low and the intra-cluster similarity is high.

5. Classification-—Classification analyzes a set of training data and constructs a model for each class based on the features in the data. A decision tree or a set of classification rules is generated by such a classification process, which can be used for better understanding of each class in the database and for classification of future data.

6. Prediction—This function predicts the possible value of some missing data or the data distribution of certain attributes in a set of objects. It involves the finding of the set of attributes relevant to the attribute of interest and predicting the value distribution based on the set of data similar to the selected objects.

Many intelligent systems employ numeric degrees of belief supplied by the users to make decisions (Pearl, 1988; Gottinger and Weimann, 1995). A relation between two data attributes is better described in quantitative expressions. Bayesian network (also called belief networks or causal networks) has been a commonly applied model for describing the relations among a set of propositions in a Bayesian space (Ras, Skowron *et al.*, 1997). A Bayesian network is usually expressed as a directed acyclic graph (DAG) whose nodes represent random variables and whose edges represent probabilistic dependencies. Under a causal interpretation, the edges represent direct causal influences. These influences are quantified by the conditional probability distribution values associated with each node. To derive the probability of a node, probabilities of their parent nodes, as well as the conditional probability distribution functions on their connecting edges are computed.

Researchers have studied the use of domain knowledge to construct a Bayesian network that is tailored to particular instances of database records (Cooper and Herskovits, 1992; Heckerman, 1996; Lee and Abbott, 2003). Techniques have been developed to construct Bayesian networks using variously available information. The attributes of relevant data objects and their probabilities to be explored from databases are propagated in the network. The resulting attribute relations extracted from database exploration can also be used to refine the network. After a goal-oriented exploration of information and extraction of causal relations on the data records, a portion of the network will be updated. The resulting Bayesian network may either be reported to the user or manipulated again for further knowledge discovery processes. The Bayesian network can also be used for direct manipulation of the relevant data attributes and their relationships in the knowledge patterns extracted, and in due course updating the Bayesian networks according to the extracted knowledge.

In DCA, we use a graphical language to describe the Bayesian network representation. A graphical language L is a triple defined as the following:

$L=<N,E,P>$

where, N is a set of nodes, E a set of edge tokens, and P a set of triples indicating the relations between members in N, $P=\{p: p=<n_1, n_2,e>\}$; where n_1, $n_2 \in N$, $e \in E$. The nodes can be distinguished in two kinds, real nodes and virtual nodes. Real nodes describe the objective entities such as the data attributes in data records. The virtual nodes describe the goals or sub-goals of knowledge discovery. The edges can also be distinguished in two kinds, directed edges and non-directed edge. Each edge connects exactly two nodes and represents the relation between these two nodes. The directed edges represent a bounded (deterministic) relationship between two nodes. While the non-directed edges represent the unbounded relationship between two nodes. Unbounded relationship means that there exist relations between the two nodes, but the relationships have not been determined yet. Two unconnected nodes have no relation between them.

As an example, the following L is used to describe the Bayesian network structure shown in Figure 3.7. We use regular font letters for real nodes and italic font letters for virtual nodes specifically.

$L= <N, E, P>$

$N=\{a, b, c, d, \mathrm{e}, \mathrm{f}, \mathrm{g}, \mathrm{h}, \mathrm{i}, \mathrm{j}, \mathrm{k}\}$

$E=\{\leftarrow, \rightarrow, —\}$

$P=\{<b, a, \rightarrow>, <d, a, \rightarrow>, <d, c, \leftarrow>, <e, b, \rightarrow>, <e, f, —>, <b, \mathrm{f}, \leftarrow>, <g, h, —>, <h, c, \rightarrow>, <h, i, —>, <\mathrm{i}, d, \rightarrow>, <\mathrm{j}, d, \rightarrow>, <\mathrm{k}, d, \rightarrow>\}$

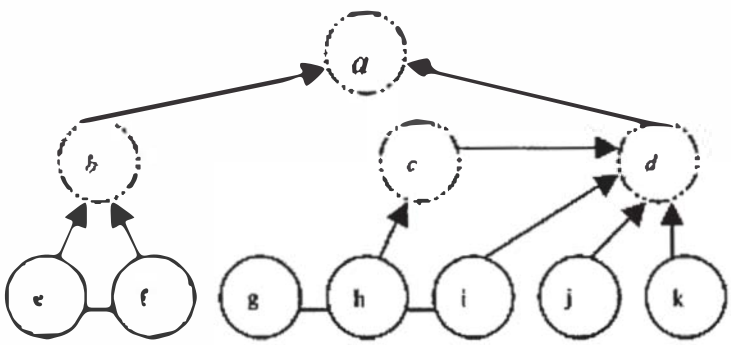

Figure 3.7 Examples of Bayesian network representation of causal relations

In the Bayesian network representation, there always exists at least one virtual node that has no direct edge pointing to it, for example node *a* in Figure 3.7. We call this node a goal (or main problem). The virtual nodes pointing to this root node are called sub-goals (or sub-problems). It is possible that there are more than one level of subgoals (sub-problems). In Figure 3.7 the sub-goals of *a* are *b* and *d*, and the sub-goal of *d* is *c*. The upper level goals can be solved when their sub-goals have been solved. In the graph representation, we use arrows represent the causal relations determined between two nodes and non-arrow line the causal relation not determined yet. The direction of arrow indicates the causal direction. The weight of the line indicates, in terms of probabilities, how strong the relation is.

To properly generate database queries from the Bayesian network, a network decomposition process is invoked. One purpose of this process is to reduce the complexity involved in the dependency representations for the goal and sub-goals. In the decomposition process, the node connections in Bayesian network relevant to the database attributes are identified. A structural analysis subsequently traces the connections and influences of the sub-goals in the network. The decomposition process will extract a goal-related sub-network from the Bayesian network. This sub-net is then converted to a goal-related tree by a net-to-tree conversion procedure. The constructed goal-tree is in a format suitable for converting back to the Bayesian networks so that the Bayesian network can be updated using the extracted knowledge from the database explorations.

The computational process of attribute exploration using the Bayesian network representation consists of the following steps.

1. Identify an ultimate goal for an attribute exploration process

2. Generate initial Bayesian network (BN), mark the ultimate goal as the current goal and put it into a goal stack.

3. Until no goals remaining in the BN not explored

- Identify a sub-goal node in the Bayesian network with link to the current goal
- Generate a conceptual query from attributes contained in the sub-goal
- Convert conceptual query to actual query using database schema of the database
- Retrieve information from database via the interactive interface
- Extract cross-reference threads from the retrieved data record
- Map the threads back to the Bayesian network
- Locate goals and sub-goals related to the threads
- Incorporate the threads to Bayesian network goals not explored yet

• Back to step 3

4. Report the resulting BN as the KP extracted.

3.2.2.2 The Query Agent (QA)

QA should have the ability to monitor, communicate, and collaborate with other agents, react to various requests, as well as assign tasks to proper DCAs. The responsibility of the QA is mainly focus on providing the optimum solutions to the user based on the information collected from the DCA. The functions of QA include:

- interpret the questions asked by the end users
- issue the appropriate request to the respective DCA for information
- gather information and collate them for decision support
- deliver an appropriate answer to the end users

Figure 3.8 shows the architecture of QA, which consists of three modules: Interface Module (to end users), Query Analysis Module, and Communication Module with DCA. Interface Module and Communication Module manage the input and output of QA. Query Analysis Module is the main components of QA, which is used to transform the user's demand into an accessible query form for DCA, and collate the data retrieved by DCA for the answer to the user. The main components of Query Analysis Module are described in details next.

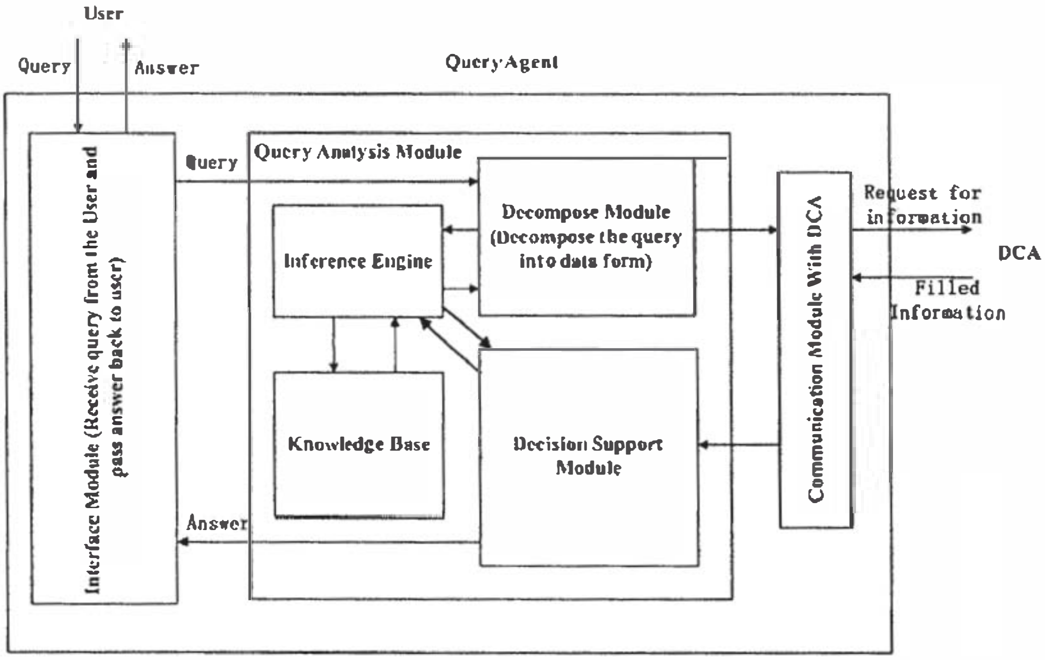

Figure 3.8 Architecture of Query Agent

First of all we need to analyze and interpret the content of the query from the user. Base on the query, the agent has to identify what type of information is needed in order to resolve this query. The query needs to be decomposed into information request related to the N types of information from DCA. This process is supported by the decompose module which decomposes the query into a data form. The determination of the content in the data form can be supported by a knowledge base having the set of rules and the selection of the particular rules can be performed by a rule based inference engine. After the data form is defined, it can be submitted to the corresponding DCA to search for the information that is needed in the form. After collecting the data back from different DCA, the QA attempts to find an optimal solution to the particular query. The decision support module is responsible for selecting the relevant rule stored in the knowledge base through the inference engine based on the particular constraint defined by the query. After the suitable rule is selected, the best solution can be calculated based on the information collected through the DCAs. The decision support module is also responsible for knowledge update.

In QA, the knowledge base contains a rule base. In the rule base, the specialized domain knowledge is presented in the form of *if <antecedent clauses> then <consequent clauses>* statements. If the antecedent clauses are true, then the consequent clauses are true. The rules in the rule base are triggered if their antecedents match. All rules must be coded in IF-THEN-ELSE statements for execution on computers and stored in the knowledge base. Upon receiving the known facts from the DCA, the QA needs to infer and explain the situation, and subsequently trigger actions or alternatives for decision makers.

Knowledge presentation plays an important role in knowledge reasoning. A well-designed knowledge presentation will affect the performance of an agent. In the past decades, many schemes of knowledge-based decision support systems, including production rules, predicate logic, semantic networks, frames, scripts, and decision trees, have been developed. However, production rules are most commonly used in knowledge presentation. Generally, knowledge-based decision support systems must have the following characteristics:

1. Representational adequacy: this is the ability to professionally describe all knowledge and to comply with all knowledge in a knowledge base.

2. Inferential adequacy: it can inference new rules from some given rules and easily build a new structure.

3. Inferential efficiency: this is the ability to efficiently reason, quickly execute, and get conclusions.

4. Acquisitioned efficiency: knowledge must be easily accessed.

As stated above, rules are presented as a set of if <antecedent clauses> then <consequent clauses> statements. Basically, there are two types of reasoning: backward chaining and forward chaining. Forward chaining is a data-driven approach; it starts from a basic idea and then tries to draw conclusions. It first checks the IF part of IF-THEN rules to find out whether the antecedent clauses of the rule is matched. As each rule is tested, the program can infer one or more conclusions. Conversely, backward chaining is a goal-driven method. It starts with a goal to be verified as either true or false. It then examines all the THEN parts of IF-THEN rules. A rule that contains this goal in its consequent clause will be checked to determine whether its antecedent clause is true. If it fails, the program searches for another rule whose conclusion is the same as that of the first one. This process continues until all possibilities have been tested. Our approach makes use of forward chaining.

Figure 3.9 shows an example of a hierarchal structure for rule presentation. For instance, we organize the knowledge base into five groups: production, inventory, sales, finance, and personnel. In order to have more space to accumulate the appending rules, the starting number begins from 100, 200, 300, 400, and 500, respectively. To clarify the forward chaining procedures, several example rules in the knowledge base are shown below:

Rule 100: if Condition="Production" then goes to Rule 101

Rule 101: if Quantity_per_month<Month_goal then Situation= "Warning"

Rule 102: if Accumulated_quantity_per_month <30% of Total_goal then Action= "Ask the production manager to find reasons."

Rule 103: if Defector_rate≥ Average_defector_rate then Action="Improve manufacturing drawbacks or techniques"

Rule 104: if Reason="Bad skills" then Action="Improve training"

Rule 105: if Reason="Low capacity" then Action="Overtime"

Rule 106: if Reason="the leak of materials" then Action="Emergency purchase and show Warning"

Rule 107: if Situation="Warning" then Action="Keep tracking"

Rule 200: if Condition="Inventory" then go to Rule 201

Rule 201: if Inventory_level > Average_inventory_level then Action="Reduce inventory"

Rule 202: if Action="Reduce inventory" then Action1 ="30% off discount for the unused parts"

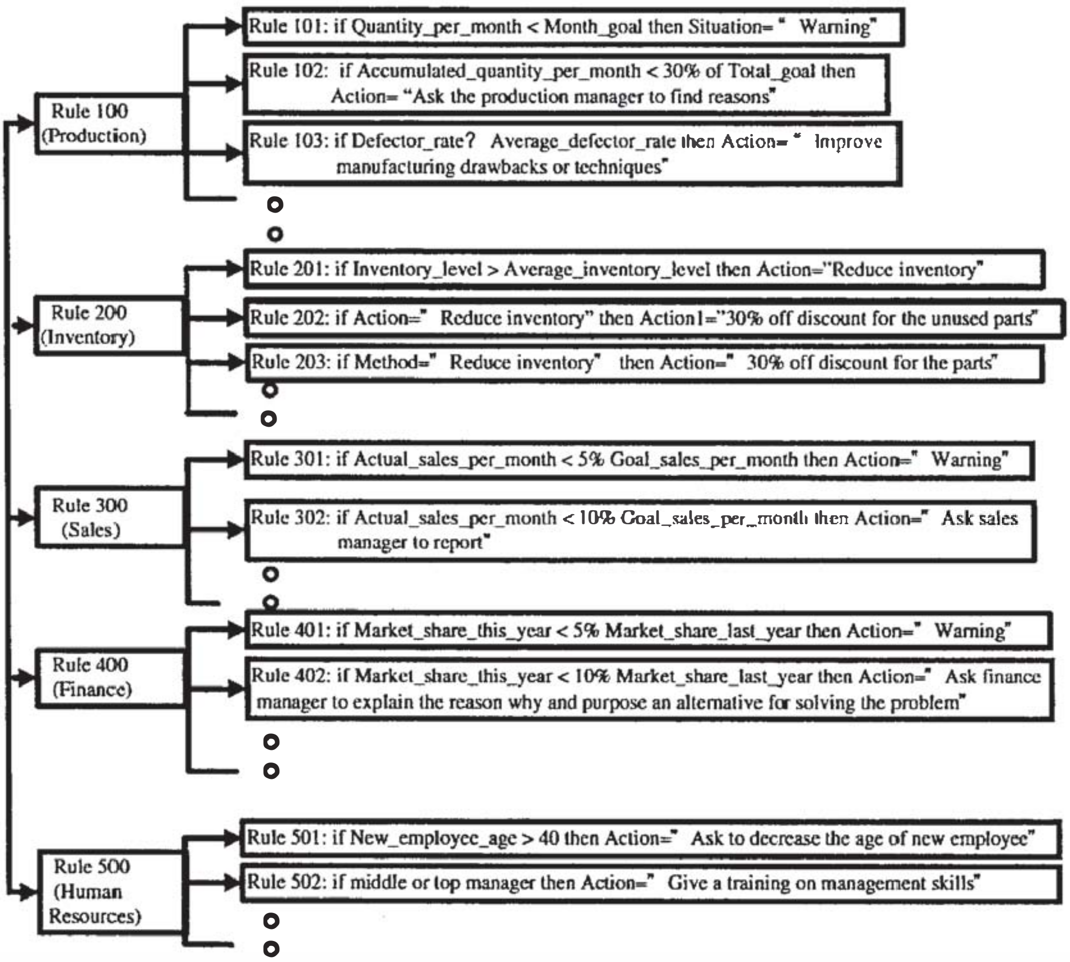

Figure 3.9 A hierarchal structure of rules

The following is an example involving a production decision by using the forward chaining procedures. In forward chaining, we start with known facts and acquire new facts using rules that have known facts on their IF sides. When the condition in the IF part of the statement is satisfied (is true), then the rule will be fired and the conclusion in the THEN part will be added to the assertion base. Assume that the condition is equal to production and the goal for this month is to produce 150 microwaves, and the production quantity of this month is 120 microwaves.

Step 1: Because the condition in the IF part is equal to production, according to Rule 100, QA points to Rule 101.

Step 2: Next, QA selects Rule 101 and gets the facts from the DCA results to reason.

Step 3: Now, Rule 101 has been fired since the value of quantity is less than the goal for this month.

Step 4: Using Rule 107, QA then keeps monitoring the progress of production because the situation is equal to warning.

During the reasoning period, when the company's goal is adjusted, QA can use the forward chaining method shown above to automatically infer again without revising the program. Therefore, the mechanism is able to easily and quickly respond to changes in the rule base.

3.2.2.3 The Enjoin Agent (EA)

The functions of the EA are similar to QA. The responsibility of an EA is to:

- analyze a command from the end user
- issue the appropriate request to the respective DCA for information
- gather information and collate them
- send the collated instructions to the designated person

The difference between the architectures of QA and EA is their input and output. The output of EA is the instructions to the person in charge, not the manager who issues the command. Therefore, EA also need to get the contact methods of the designated person from DCA. The architecture of the EA is shown in Figure 3.10.

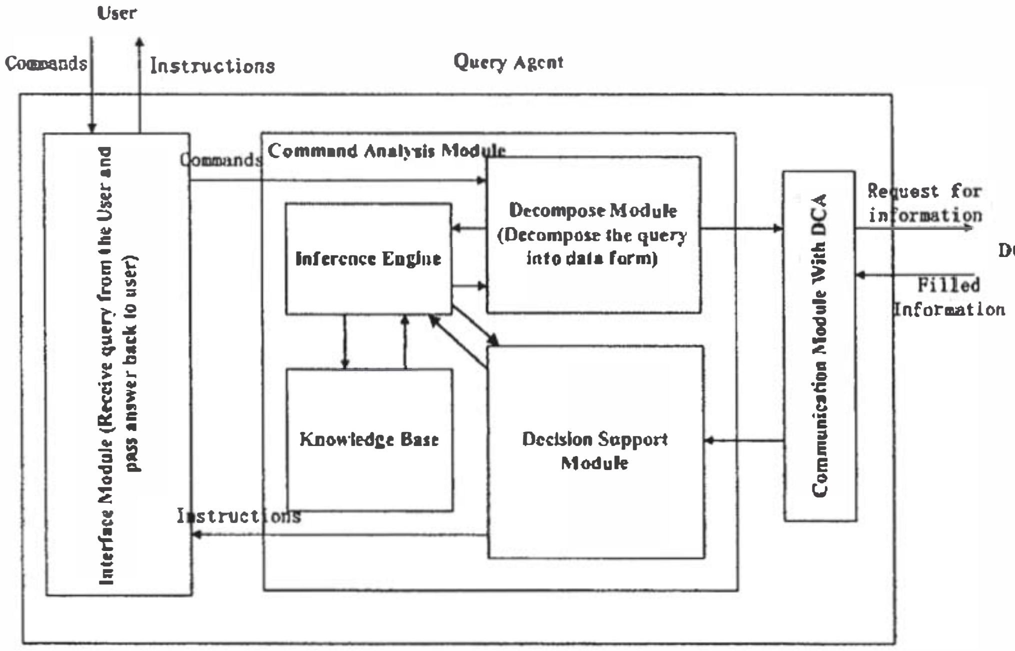

Figure 3.10 Architecture of Enjoin Agent

Our proposed Enjoin Agent is supported by the rule based inference engine and the knowledge base which is for decomposition of the Query into data form as well as selection of the mathematical model for the optimization process. Same as query agent, the first step is to analyze and interpret the content of the command from the user. Base on the command, the agent has to identify what type of information is needed in order to resolve this command. The command needs to be decomposed into information request related to the N types of information from DCA. This process is supported by the decompose module which decomposes the command into a data form. The determination of the content in the data form can be supported by a knowledge base having the set of rules and the selection of the particular rules can be performed by a rule based inference engine. After the data form is defined, it can be submitted to the corresponding DCA to search for the information that is needed in the form. After collecting the data back from different DCA, the EA attempts to collate instructions and distribute them to the designated person(s). The decision support module is responsible for selecting the relevant rule stored in the knowledge base through the inference engine based on the particular constraint defined by the command. After the suitable rule is selected, the instruction can be formed based on the information collected through the DCAs. The decision support module is also responsible for knowledge update.

Knowledge representation is very important. CBR approach alone is not adequate to solve the problem in case the number of rules or cases is huge. A frame type representation is needed for proper management and representation of the cases. In such case an integrated conceptual graph together with frame type rule based approach need to be used to enhance the process of situation mapping and fast selection of the appropriate cases. Frame type structure help organize the problem closer to an object oriented approach. It provides a much better representation scheme of the knowledge and problem. We assume that command input is already in a pre-defined structured. We need to translate this pre-defined structure into frame typed CBR architecture for the selection of cases. The SITUATIONS frame is used to match the situation with the corresponding case. Select the best suitable case base on the fitting ratio and the garbage ratio and measure the degree of coverage of the cases stored in the case base and the degree of dissimilarity of that case. The fitting ratio is calculated between the situation frames of the new problem and those of the cases stored in the case base. The fitting ratio and the garbage ratio are used in a cumulative way. If a retrieved case is not enough for the explanation of the characteristics of new problem, several cases are selected in sequence according to the fitting ratio value. If all the situation frames of the new problem are explained by those of the retrieved cases, the new problem is said to be covered by the retrieved cases. Conceptual Graph – Frames – Rules is a unified representative scheme combines the feature of logic and network based languages along with rules that combines the features of Conceptual Graphs (logic and network based), frames (network based), and rules. CGs are implemented on top of the frame system. The formal structure of a frame is given by the regular expression: ((frame-name > ((slot-name> ((facet-name>(value)*)* (demon)*)*). The combination of CG frames and rules allow us to represent shallow and deep knowledge explicitly. This representation scheme retains the modularity while permitting an explicit structural description of the semantics of the problem domain.

Chapter 4 The Intelligence Distribution Strategy

4.1 Problem Description and Formulation

4.1.1 Problem Description

In a mobile enterprise, a set of information tasks is identified by users. To accomplish all tasks, each agent in our MMAP was designed to possess specific intelligence (called functions) such as the ability to check stock level, check past sales record, and so on. The performance of agents is determined by a combination of intelligences for each agent and the appointment of tasks to intelligences in the agent. In this section we attack the issue of designing and optimizing MMAP to offer helpful services.

A number of pertinent functions might require related resources (e.g., database). Each task contains one or more information units, while different tasks could contain the same information units. An information unit is the smallest processing piece, and each must be processed by only one function in an agent. The completion rate of obtained information (CRI) of each unit performed by the related function in a different agent would be different as well. Hence, based on all tasks' contents, the set of functions must be assigned to the agents. Additionally, the allocation of information units to functions must be optimized in each MMAP agent to achieve the maximum total CRI.

The following query example is given for interpretation of the definition of intelligence. There are three types of task request: Profit, Delivery and

Outsource. The structure of the <<Query Frame>> is shown as the following. The frames are identified by the part number of the product with the situation as the conditions.

{{Part Number: part number

Task : Profit

Situation : [Price][Quality][Lead Time][Cost]

Actions : Sales Form [Past Sales Record], part number

Warehouse Form [Stock Level][Production Lead Time][Outsource Possibility][Outsource Cost], part number

Financial Form [Financial situation], part number

}}

{{Part Number: part number

Task : Delivery

Situation : [Price][Quality][Cost]

Actions : Sales Form [Past Sales Record], part number

Warehouse Form [Stock Level][Production Lead Time][Outsource Possibility][Outsource Cost], part number

Financial Form [Financial situation], part number

}}

{{Part Number: part number

Task : Outsource

Situation : [Price][Quality][Lead Time][Cost]

Actions : Sales Form [Past Sales Record], part number

Warehouse Form [Stock Level][Production Lead Time][Outsource Possibility][Outsource Cost], part number

Financial Form [Financial situation], part number

}}

After selecting the optimization method, it will be represented in a frame structure <<Optimization Frame>>:

Optimization Frame:

{{Name: O1

Situation: [Price][Quality][Lead Time][Cost][Past Sales Record] [Stock Level][Production Lead Time][Outsource Possibility][Outsource Cost] [Financial situation]

Input variables from situation: [Vpr][Vqt][Vlt][Vco]

Filled the information: [Fps][Fsl][Fpl][Fop][Foc][Ffi]

Methodology: EOI

Deliverables from optimization: Profit

Make in house / outsource

Production lead time / Outsource lead time & Cost

Financial Impact to company

Volume commitment

}}

[Vpr] means the value of the price that is being input from the original query

[Vqt] means the value of the quantity that is being input from the original query

[Vlt] means the value of the lead time that is being input from the original query

[Vco] means the value of the cost that is being input from the original query

[Fps] represents the Filled information from the DCA about the past sales record.

[Fsl] represents the Filled information from the DCA about the stock level.

[Fps] represents the Filled information from the DCA about the production lead time.

[Fop] represents the Filled information from the DCA about the outsource opportunity.

4.1.2 Assumptions

The following assumptions are made for the addressed problem:

1. Each agent contains one or more functions.

2. Some functions are assigned only to specific agents because certain resources required by the functions can only be visited by specific agents.
3. Some functions conflicts that cannot be assigned to a single agent.
4. Each type of function may be multiple and assigned to different agents.
5. Each function in each agent must be utilized.
6. One function would process one or more information units in tasks.
7. Each task's content is fixed and known.
8. The process time of units is neglected.
9. Each unit's CRI processed by each function in each agent is fixed and known.
10. Machine breakdown is not considered.
11. Resource is able to meet the demand of functions.

4.1.3 List of Symbols

To formulate the MMAP model, the following notations are introduced:

Indices:

i = index for tasks ($i = 1, \cdots, I$)

j = index for information units in tasks i ($j = 1, \cdots, J_i$)

k = index for resources ($k = 1, \cdots, K$)

m = index for agents ($m = 1,\ldots,M$)

n = index for functions ($n = 1,\cdots,N$)

Parameters:

N= function set;

N' = a subset of the task set N, one element in N' represents a function requiring some resource;

$$R_{kn} = \begin{cases} 1, & \text{if resource } k \text{ is required by function } n \\ 0, & \text{otherwise} \end{cases};$$

$$S_{km} = \begin{cases} 1, & \text{if resource } k \text{ can be visited by agent } m \\ 0, & \text{otherwise} \end{cases};$$

$$C_{n_1 n_2} = \begin{cases} 1, & \text{if function } n_1 \text{ and function } n_2 \text{ are conflict} \\ 0, & \text{otherwise} \end{cases};$$

α_{mnj} = CRI of unit j processed by function n in agent m;

$$\beta_{ij} = \begin{cases} 1, & \text{if task } i \text{ contains information unit } j \\ 0, & \text{otherwise} \end{cases}$$

Decision variables:

$$x_{mn} = \begin{cases} 1, & \text{if function } n \text{ is to be assigned to agent } m \\ 0, & \text{otherwise} \end{cases}$$

$$y_{mnj} = \begin{cases} 1, & \text{if unit } j \text{ is processed by function } n \text{ in agent } m \\ 0, & \text{otherwise} \end{cases}$$

4.1.4 Objective Function and Constraints

Tbe objective function and constraints are considered as follows:

$$\text{Maximize} \quad \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{i=1}^{I}\sum_{j=1}^{J_i}\alpha_{mnj}\beta_{ij}y_{mnj} \qquad (1)$$

Subject to: $$\sum_{n=1}^{N} x_{mn} \geq 1, \quad \forall m \tag{2}$$

$$\sum_{m=1}^{M} x_{mn} \geq 1, \quad \forall n \tag{3}$$

$$R_{kn} x_{mn} \leq S_{km} \qquad \forall k, m, \ n \in N' \tag{4}$$

$$x_{mn_1} x_{mn_2} C_{n_1 n_2} = 0, \qquad \forall m, n_1, n_2, \ n_1 \neq n_2 \tag{5}$$

$$y_{mnj} \leq \lceil \alpha_{mnj} \rceil x_{mn}, \quad \forall m, n, j \tag{6}$$

$$\sum_{i=1}^{I} \sum_{j=1}^{J_i} \beta_{ij} y_{mnj} \geq 1, \quad \forall m, n \tag{7}$$

$$\sum_{m=1}^{M} \sum_{n=1}^{N} y_{mnj} = 1, \quad \forall j \tag{8}$$

$$x_{mn}, y_{mnj} \in \{0, \ 1\}, \quad \forall m, n, j \tag{9}$$

where the objective function given in equation (1) maximizes the MMAP's total CRI, which is equal to the sum of the CRI of each unit processed by each function in each agent. Constraint (2) ensures that each agent contains one or more functions. Constraint (3) ensures that each type of function must be assigned to agents. Constraint (4) ensures select functions are assigned only to specific agents. Constraint (5) specifies functions are conflicts that cannot be assigned to any agent. Constraint (6) ensures that, when unit j is assigned to agent m, there must be a function n processing unit j in agent m. Constraint (7) ensures that each function in each agent must be utilized. Constraint (8) ensures that each unit must be processed by only one function in an agent. Constraint (9) is the integral requirement.

The MMAP model is a nonlinear 0-1 programming model. Solving it by means of traditional methods is difficult. As the number of decision variables and constraints increase substantially with the increased number of tasks, units, functions, and agents, the computation becomes more complicated and time-consuming. In this case, a meta-heuristics aiming for a near optimal solution with reasonable computation time may be more appropriate and can be easily accepted by enterprises. Hence, an efficient scatter search approach is developed in the next section to solve the problem.

4.2 A Scatter Search Approach

Scatter search was first introduced by Glover (1977) as a heuristic for integer programming. It is an established meta-heuristics that has received renewed attention over the last decade. From the meta-heuristics classification, the scatter search may be viewed as an evolutionary (also called population-based) algorithm that constructs solutions by combining others. This methodology's goal is to enable the implementation of solution procedures that can derive new solutions from combined elements (Glover, 1998).

The scatter search methodology is very flexible, and each of its elements can be implemented in a variety of ways. Moreover, the algorithm incorporates procedures based on different strategies, such as diversification, local search, tabu search, or path relinking (Haq, Saravanan *et al.*, 2007). Hence, scatter search has recently been applied successfully to solve various complex problems (Jiang and Yung; Jennings, 2000; Laguna and Marti, 2002; Lea, Gupta *et al.*, 2005).

4.2.1 Generating Initial Solutions

To solve the model MMAP through the scatter search approach, the first task is to present a solution for the addressed problem (Mart, Laguna *et al.*, 2006). The solution encoding for the proposed model involves the 0-1 decision variable x_{mn} and y_{mnj}, enabling an initial solutions generation heuristic that can satisfy parts of the constraints.

Figure 4.1 illustrates the encoding structure of a solution. This representation consists of two segments. The first segment, labelled "Agents," represents the assignment of functions to various agents. The second segment, labelled "Information Units," represents the assignment of units to various agents. In Figure 4.1, x_{mn} in the Agents segment and m_j in the Information Units segment are presented in detail. The element x_{mn} in the representation assumes a 0-1 integer value. If function n is assigned to agent m, then x_{mn} is one; otherwise, x_{mn} is zero. In the Information Units segment, m_j is the agent number; this means that unit j assigned to agent m. Based on the representation, a generation method of initial solutions for the addressed problem is presented in Algorithm 1.

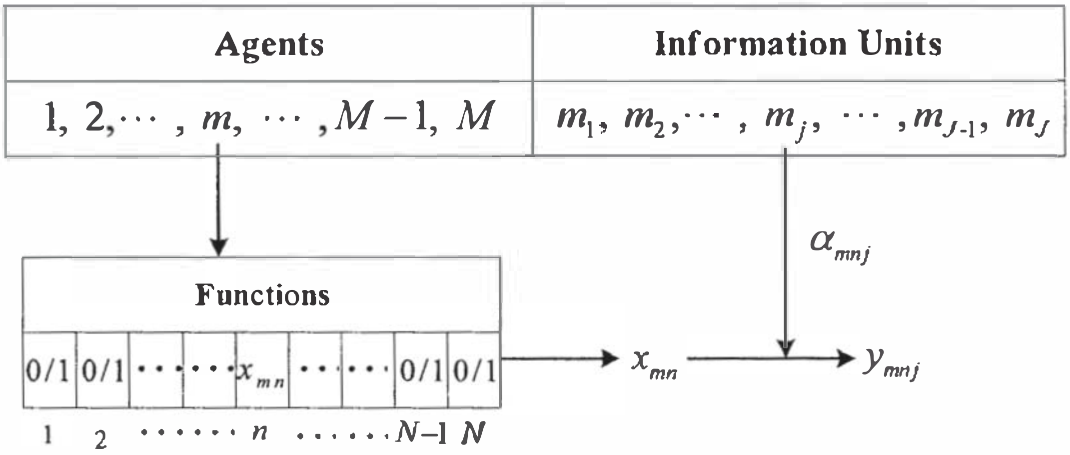

Figure 4.1 Encoding structure of a solution

Algorithm 1 (Encoding method of the problem MMAP)

Step 1: Let K be a universal set of resource, where each element represents a kind of resource. Let set F_1 be a universal set of functions. Let set A_1 be a universal set of agents.

(Certain functions are only assigned to specific agents.)

Step 2: Choose resource k randomly, $k \in K$, and determine a function set F_2 and an agent set A_2 based on the resource k. All functions requiring resource k are included in set F_2 and all agents visiting resource k are included in set A_2.

Step 3: Choose function f and agent a randomly, $f \in F_2$, $a \in A_2$.

Step 4: If function f is in conflict with others in agent a, proceed to Step 3. Otherwise, assign function f to agent a and delete f from set F_2 and set F_1 (if set F_1 includes f), respectively.

Step 5: If set $F_2 = \varnothing$, delete k from set K. Otherwise, proceed to Step 3.

Step 6: If set $K \neq \varnothing$, proceed to Step 2.

(Each type of function must be assigned to agents.)

Step 7: Choose function f and agent a randomly, $f \in F_1$, $a \in A_1$.

Step 8: If function f is in conflict with others in agent a, proceed to Step 7.

Otherwise, assign function f to agent a and delete f from set F_1.

(Each agent contains one or more functions.)

Step 9: Find out the agent set A_3, where each element represents an empty agent.

Step 10: Choose agent a randomly, $a \in A_3$.

Step 11: Choose agent b randomly, which includes two or more functions, $b \in A_1$.

Step 12: Move a function from agent b to agent a, and delete a from set A_3.

Step 13: If set $A_3 \neq \varnothing$, proceed to Step 10.

Step 14: Choose information units randomly and assign these to agents. Stop.

According to the representation and Algorithm 1, the value of decision variable x_{mn} can be read directly from the encoding. Considering the statement, "an information unit is the smallest processing piece, and each unit must be processed by only one function in an agent", y_{mnj} can be obtained based on x_{mn}, m_j and α_{mnj}:

$$y_{mnj} = \begin{cases} 1, & x_{mn} = 1,\ m_j = m,\ \alpha_{mnj} > 0 \\ 0, & \text{otherwise} \end{cases} \qquad (10)$$

The generated initial solutions may violate constraints given in equation (6-8). Hence, the initial solutions require transformation into legitimate solutions via the repair method presented in the section below.

4.2.2 Generating Initial Reference Set

The reference set is a collection of both high-quality and diverse solutions used to generate new ones by applying the combination method (Pacheco, 2005). The reference set is constructed in the initial step, and it consists of b_1 high-quality solutions and b_2 diverse solutions from the diverse initial solutions. The reference set's size is denoted by $b=b_1+b_2$. The selection of solutions that will be included in the reference set, considering their quality as well as diversity, will create a high impact on the quality of new solutions generated by the combination method. The b_1 high-quality solutions are selected based on the objective function values, that is, the solutions with lesser total tardiness penalty costs are selected as high-quality solutions. To determine b_2 diverse solutions, a diversity measure must be defined as follows (Burcu et al., 2006):

$$d(\mathbf{s}_1,\mathbf{s}_2)=\sum_{i=1}^{n}\left|s_1^i-s_2^i\right| \tag{11}$$

where $d(\mathbf{s}_1,\mathbf{s}_2)$ connotes the minimum distance between two solutions $\mathbf{s}_1=(s_1^1,\cdots,s_1^n)$ and $\mathbf{s}_2=(s_1^1,\cdots,s_1^n)$.

After high-quality solutions in the reference set have been included, the minimum distances are computed for each solution in the initial population. Subsequently, the solution with maximum distance is selected. This solution is added to the reference set and deleted from the initial population. Repeat the process until b_2 diverse solutions are achieved. The resulting reference set contains b_1 high-quality solutions and b_2 diverse solutions.

4.2.3 Generating Subsets

The subset generation method operates on the reference set to produce a subset of its solutions as basis for creating combined solutions. There are several systematic ways to generate these subsets. The number of subsets has an impact on the algorithm's overall duration. To decrease the overall duration and simplify the subset generation method, a procedure for generating subsets of reference solutions is adopted (Petit-Roze and Strugeon, 2006). The strategy expands pairs into larger-sized subsets while controlling the total number of subsets to be generated. It thus avoids the extreme type of process that creates all subsets of size 2, then size 3, and so on until reaching size b-1 and, finally, the entire reference set. The four types of subsets are presented as follows:

- Subset type 1: all 2-element subsets
- Subset type 2: 3-element subsets derived from the 2-element subsets by augmenting each 2-element subset to include the best solution not in this subset
- Subset type 3: 4-element subsets derived from the 3-element subsets by augmenting each 3-element subset to include the best solutions not in this subset
- Subset type 4: the subsets consisting of the best i elements, for $i = 5$ to b

4.2.4 Combining Solutions

The combination method utilizes the solutions' objective values as coefficients in the calculation (Prabhaharan, Ramesh *et al.*, 2007). The combination method is a problem-related mechanism as it is deeply linked to the solution representation. Since the representation is composed of the Agents and Information Units segments, the solution combination method must be applied to the two segments respectively.

Let $\mathbf{S}$ be a subset under consideration for generating a new solution $\mathbf{u}$ including a segment information, and assume that $\mathbf{S}$ contains the solutions $\{\mathbf{s}_1,\cdots,\mathbf{s}_n\}$ with objective values $\{f(\mathbf{s}_1),\cdots,f(\mathbf{s}_n)\}$. To determine the value of each element u_i in $\mathbf{u}$, we use the following formula:

$$u_i = \frac{\sum_{k=1}^{n} s_k^i \times f(\mathbf{s}_k)}{\sum_{k=1}^{n} f(\mathbf{s}_k)}, \quad u_i \in \mathbf{u} \tag{12}$$

where s_k^i is the ith element in solution $\mathbf{s}_k$, and $f(\mathbf{s}_k)$ is the total CRI. Since the problem MMAP is a maximum problem, $f(\mathbf{s}_k)$ is used as a multiplier so that a solution with a higher objective value receives higher weight. Evidently, an element of a combined solution can be a fractional value. To obtain a legitimate solution, u_i is rounded up to its closest integer value.

4.2.5 Improving Solutions

The improvement method's goal is to transfonn a trial solution into one or more enhanced trial solutions. Neither the input nor the output solutions need to be feasible, though the output solutions are typically feasible (Symeonidis, Kehagias *et al.*, 2003). A scatter search procedure can be implemented without the improvement method, but it is essential for high-quality solutions.

This paper adopts a simple and effective improvement method based on the neighborhood search (Sheremetov, Contreras *et al.*, 2004; Rogers, David *et al.*, 2007). The improvement method maintains the same sequence information of the original solution by merely swapping two randomly selected elements. First, the two elements u_i and u_j are randomly selected from a solution $\mathbf{u}$. Second, the new solution will be generated by swapping the positions of elements u_i and u_j after the improvement method. Similar to the combination method, the improvement method must be applied to the two segments as well.

4.2.6 Repairing Solutions

Within the development of the proposed scatter search approach, trial solutions may violate constraints given in equation (2-8). Hence, a repair method is proposed to transform solutions violating the constraints into legitimate solutions. The repair method contains five heuristic procedures to repair illegitimate solutions.

The first repair heuristic procedure is employed to correct the violation of constraints (6-8). To perform this adjustment operation, the heuristic must first determine the units in each agent based on the information units segment; then determine the idle functions in each agent, and remove the functions from the related agent; or find out a unit in its agent, which is not processed by functions in the agent, and assign the related function to the agent.

The second repair heuristic procedure involves correcting the violation of constraint (5). The heuristic first determines function *n* and function *n'*, which are in conflict and assigned in agent *m*. Function *n* must then be chosen and moved to a suitable agent. The place of units processed by function *n* must be adjusted.

The third repair heuristic procedure involves correcting the violation of constraint (4). To perform this adjustment operation, the heuristic first determines resource *k*, function *n*, and agent *m* when $x_{mn} = 1$, $R_{kn} = 1$, and $S_{km} = 0$. Another agent *m'* must be chosen randomly, where each function does not conflict with function *n*. Function *n* must be moved from agent *m* to agent *m'* and the place of units processed by function *n* must be adjusted.

The next repair heuristic procedure involves correcting the violation of constraint (3). The heuristic first determines function *n*, which is not assigned and required by certain units. An agent *m* must then be chosen randomly, where each function does not conflict with function *n*, and then let $x_{mn} = 1$.

The final repair heuristic procedure involves correcting the violation of constraint (2). The heuristic first determines empty agent *m*. Agent *m'* must be chosen randomly, including two or more functions. Finally, any function must be moved from agent *m'* to agent *m* and the place of units processed by the function must be adjusted.

To obtain legitimate solutions, the five heuristic procedures are applied according to the above order. The heuristic displays randomness in behavior, making it compatible with the scatter search approach. Through the repair method, the generated solutions can satisfy constraints in the equations (2-10).

4.2.7 Update Reference Set

The reference set updating method accompanies each application of the improvement method. The update operation consists of maintaining the record of b_1 high-quality solutions and b_2 diverse solutions, where the value of b ($b=b_1+b_2$) is stated as a chosen constant, but may readily be allowed to vary. Once the new solutions have been generated, the better ones in the reference set are included while the worst are excluded from the reference set. The stopping criterion is defined by the convergence of the RefSet (Tang, Wang *et al.*, 2008) (i.e., when in any iteration no new solutions are included in the RefSet).

4.2.8 Framework and Overall Procedure of the Scatter Search for Solving MMAP

To summarize the above analysis, the framework of the proposed scatter search approach to solve problem MMAP is illustrated in Figure 4.2.

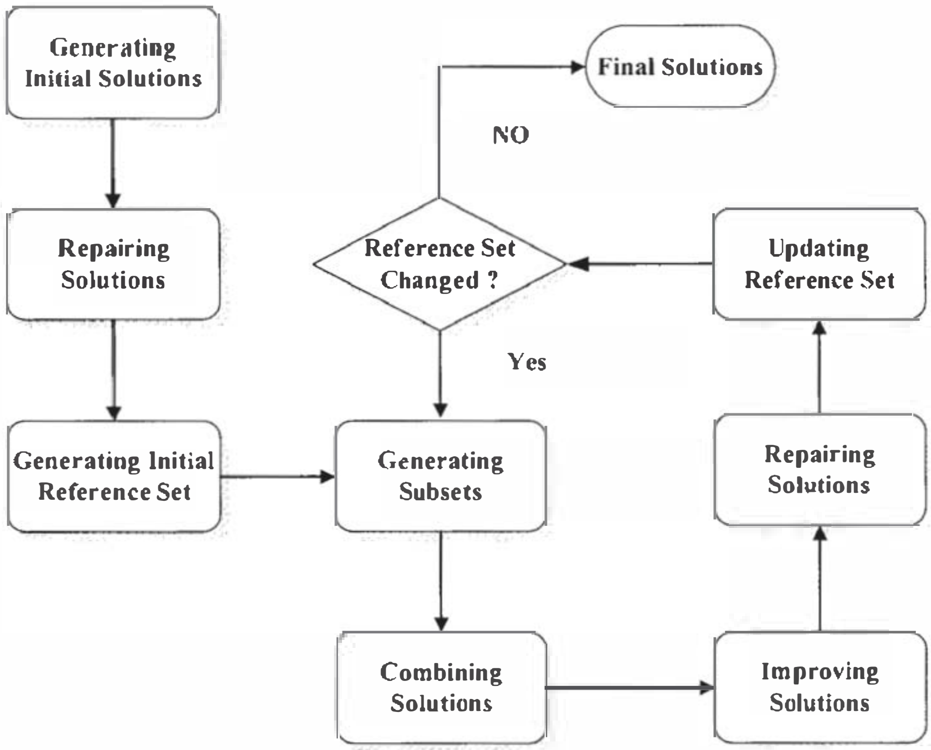

Figure 4.2 Framework of the proposed scatter search approach

4.3 Example and Simulation Results

To illustrate the MMAP model's application, an example and its simulation results are presented in this section.

An electronic appliance manufacturing company in Southern China produces over 10 different kinds of appliances such as electronic stoves, heaters, blenders, and rotisseries to name a few. The company is divided into a number of major departments: Procurement, Planning, Manufacturing, Sales, Warehouse, Financial, Quality, and so on. The different departments are dedicated to processing a number of information units in different tasks. The tasks are offered from the manager of company.

To boost market competitiveness, it is essential to design a MMAP for the company. For simplicity, 4 kinds of resources (denoted as Resource 1, Resource 2, Resource 3, and Resource 4), 12 kinds of functions (labeled F1 to F12) in 5 agents (denoted as Agent 1, Agent 2, Agent 3, Agent 4 and Agent 5), and 20 information units (labeled from U1 to U20) in 3 tasks (denoted as Task 1, Task 2 and Task 3) are selected in the example for purposes of illustration.

Tables 4.1 and 4.2 demonstrate the capability of agents visiting resources and the functional requirement for different resources, respectively. Symbol "√" represents one kind of resource that can be visited or required by an agent or a kind of function in Tables 4.1 and 4.2. Table 4.3 displays the information units included in tasks. Symbol "√" represents a task that includes one information unit. Table 4.4 shows the CRI of each unit processed by each function in agents. For example, the first number in Table 4 is "0.7093," which means that the CRI of U1 processed by F2 in Agent 1 is "0.7093." Table 4.4 demonstrates that one unit can be assigned to many agents, but the CRI of each assignment can be different.

Table 4.1 Capability of agents visiting resources

	Agents	Resource 1	Resource 2	Resource 3	Resource 4
1	Agent 1	√	√	√	
2	Agent 2		√	√	√
3	Agent 3	√	√		√
4	Agent 4		√	√	√
5	Agent 5	√		√	√

Table 4.2 Requirement of functions for different resources

	Functions	Resource 1	Resource 2	Resource 3	Resource 4
1	F1				√
2	F2		√		
3	F3				
4	F4		√		
5	F5				
6	F6				
7	F7	√			
8	F8				
9	F9			√	
10	F10				
11	F11				√
12	F12		√		

Table 4.3 Information units in tasks

Units	Tasks			Units	Tasks		
	Task 1	Task 2	Task 3		Task 1	Task 2	Task 3
U1	√		√	U11	√		
U2	√	√	√	U12		√	
U3		√	√	U13	√		√
U4	√	√		U14	√	√	
U5			√	U15		√	√
U6	√	√		U16	√		
U7	√			U17		√	√
U8			√	U18	√		
U9	√			U19		√	√
U10		√	√	U20	√		

Table 4.4 CRI of each unit processed by each function in agents

Units	Functions	Tasks				
		Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
U1	F2	0.7093	0.2362	0.1194	0.6073	0.4501
U2	F1	0.3477	0.15	0.5861	0.2621	0.0445
U3	F3	0.2278	0.4981	0.9009	0.5747	0.8452
U4	F6	0.8589	0.7856	0.5134	0.1776	0.3986
U5	F10	0.5338	0.1092	0.8258	0.3381	0.294
U6	F6	0.6407	0.3288	0.6538	0.7491	0.5832
U7	F9	0.0986	0.142	0.1683	0.1962	0.3175
U8	F7	0.1249	0.0244	0.2902	0.3175	0.6537
U9	F12	0.8147	0.9058	0.127	0.9134	0.6324
U10	F8	0.0975	0.2785	0.5469	0.9575	0.9649
U11	F6	0.1576	0.9706	0.9572	0.4854	0.8003
U12	F7	0.1419	0.4218	0.9157	0.7922	0.9595
U13	F4	0.6557	0.0357	0.8491	0.934	0.6787
U14	F10	0.7577	0.7431	0.3922	0.6555	0.1712
U15	F5	0.8235	0.6948	0.3171	0.9502	0.0344
U16	F11	0.4387	0.3816	0.7655	0.7952	0.1869
U17	F7	0.4898	0.4456	0.6463	0.7094	0.7547
U18	F11	0.4984	0.9597	0.3404	0.5853	0.2238
U19	F2	0.276	0.6797	0.6551	0.1626	0.119
U20	F6	0.7513	0.2551	0.506	0.6991	0.8909

Based on the data in Tables 4.1 to 4.4, a near optimal solution for the above mentioned problem is obtained by the proposed scatter search. The algorithm has been coded in MATLAB and run on a PC with a Pentium Dual-Core processor, two 2.5 GHz internal clock, and 2G RAM memory. The average execution time for the sample problem is 38.92 seconds. The total CRI for the problem is 24.9194. Table 4.5 presents the results of functions and information units assigned to each agent. From this, one can determine that the number of functions and units in each agent is different and this assignment can satisfy constraints in equations (2-10). For example, there are no idle agents and useless functions, and each unit can be processed by the related function in its agent.

As indicated in the results, one can observe that the assignments of functions to each agent and the appointments of information units are determined. In this manner, we can deal with the identification of function groups and units groups simultaneously, as well as design an excellent MMAP with a higher total CRI.

Table 4.5 Results of functions and information units assigned to each agent

	Agents	Functions	Units
1	Agent 1	F2, F5, F9	U1, U14, U15
2	Agent 2	F2, F6, F11, F12	U4, U9, U11, U18, U19
3	Agent 3	F1, F3, F10, F11	U2, U3, U5, U16
4	Agent 4	F4, F6, F8	U6, U10, U13
5	Agent 5	F6, F7, F9	U7, U8, U12, U17, U20

The following code is the part of the main program coded in MATLAB.

```
function [ maxSolution ] = ScatterSearch( )

% Scatter Search algorithm;

% minSolution

Parameter;    % parameter of problem and algorithm, in Parameter.m

startPointSet=InitialR();          % Randomly generate starting points

ObjSet=ObjectValue(startPointSet);      % calculate object value

RefSet=GenRef(startPointSet,ObjSet);   % Generate Initial Reference Set from the initial
solutions

i=1;

flag=0;

while i<=N_iter

%while flag==0

    % Generate SubSet from the RefSet;

    [SubSetT1,SubSetT2,SubSetT3,SubSetT4]=GenSubSet(RefSet);

    % Combining Solutions, and generate New solutions from the SubSet

    NewSolutions=ComSubSetB(SubSetT1,SubSetT2,SubSetT3,SubSetT4);

    %Improving methods

    EnhancedS=Improvement(NewSolutions); % Generate NewSolutions,Swapping

    EnhancedSet=RepairSet(EnhancedS);          % Repair Enhanced points

    ObjSet=ObjectValue(EnhancedSet);       % calculate object value

    NewRef=UpdatRef(EnhancedSet,ObjSet,RefSet);

    flag=IsequalRef(RefSet,NewRef);
```

```
    % RefSet(1,:)      % object value
    if(flag==0)
        RefSet=NewRef;
    end
    i=i+1;
end
maxSolution=RefSet(1,:);


function [ startPointSet ] = InitialR(   )
%   Generate starting solutions
Parameter;     % parameter of problem and algorithm, in Parameter.m
LengthSolution = M*N;
AgentSet=round(rand(N_Initial,LengthSolution));
UnitSet=round(1+(M-1)*rand(N_Initial,J));
PointSet=cat(2, AgentSet, UnitSet);
startPointSet=RepairSet(PointSet);
%startPointSet=Repair(PointSet);


function [ ObjSet ] = ObjectValue( startPointSet )
Parameter;     % parameter of problem and algorithm, in Parameter.m
numSet=size(startPointSet,1);
ObjSet=zeros(1,numSet);
```

```
for nums=1:numSet
    for i=1:I
        for j=1:J
            mdes=startPointSet(nums,M*N+j);
            for n=1:N
ObjSet(nums)=ObjSet(nums)+startPointSet(nums,(mdes-1)*N+n)*Amnj(mdes,n,j)*Bij(i,j);
            end
        end
    end
end


function [ RefSet ] = GenRef( startPointSet,ObjValue )
% Generate Initial Reference Set from the initial solutions
Parameter;    % parameter of problem and algorithm, in Parameter.m
LengthSolution = M*N+J;
RefSet=zeros(b1_RefSet+b2_RefSet,LengthSolution+1);
for i=1:b1_RefSet
    [maxValue,maxPos]=max(ObjValue);
    RefSet(i,1)=maxValue;
    RefSet(i,2:LengthSolution+1)=startPointSet(maxPos,:);
    ObjValue(maxPos)=Mintemp;
end
```

```
for i=1:b2_RefSet

    distPos=Distance(RefSet,startPointSet,ObjValue);

    RefSet(b1_RefSet+i,1)=ObjValue(distPos);

    RefSet(b1_RefSet+i,2:LengthSolution+1)=startPointSet(distPos,:);

    ObjValue(distPos)=Mintemp;

end
```

Chapter 5 Design and Implementation of MMAP

5.1 An Illustration of MMAP

This chapter describes the design and implementation of the MMAP in real practice. In order to show how the proposed system can form the basis of mobile enterprises, the case of an executive of a company on the move in using the system to assist his decision and to control operations in the company is used to demonstrate how the proposed system is responding to the enquiry and execute his/her instructions in an intelligent manner. The system integrates a smart phone, a PDA, or a notebook with a built-in wireless card, a wireless network, which supports 802.11a, 802.11b, and 802.11g, and two high speed servers as shown in Figure 5.1. The first server is a real-time data warehouse for data acquisition and storage. The second server has a multi-agent program for handling the user's request for information.

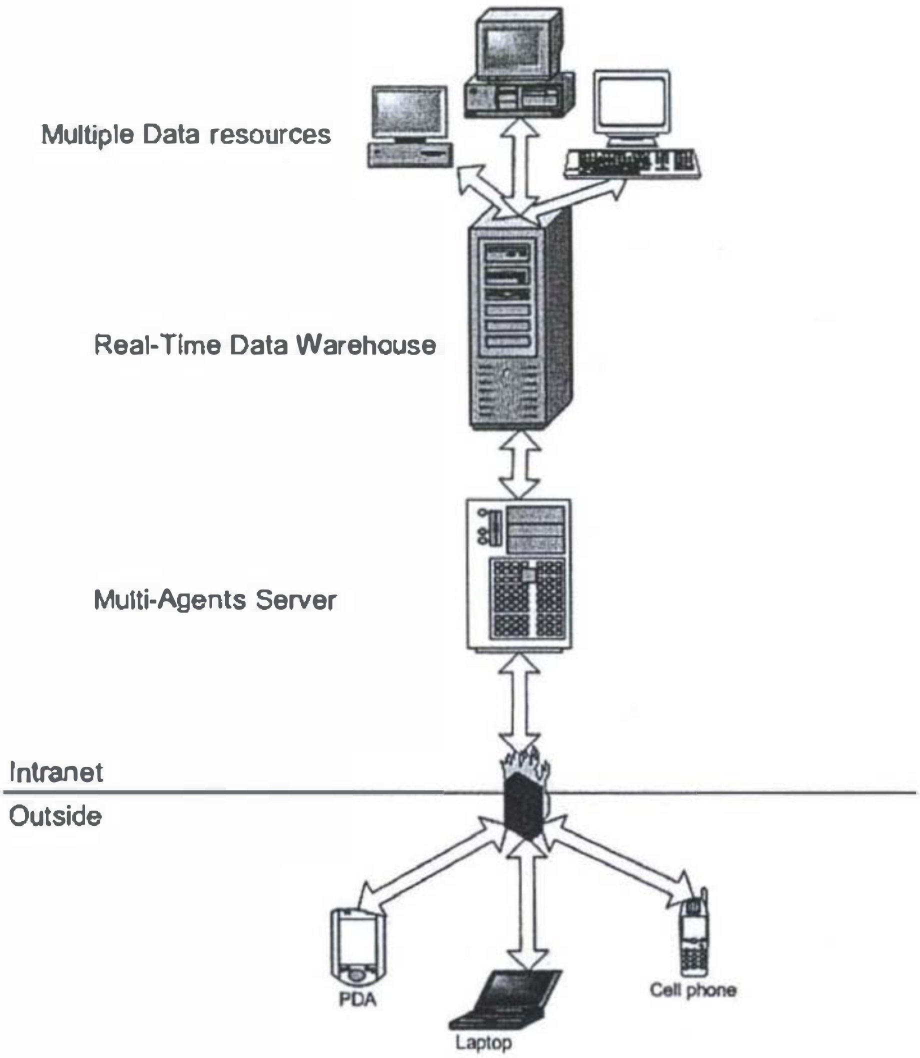

Figure 5.1 Example of Hardware Implementation in a Medium Size Printed Circuit Board Manufacturing Company

5.1.1 An Application Scenario

Take the example of a PCB (Printed Circuit Board) manufacturing company with five departments: marketing, production, financial, warehouse, and distribution. There are about 50 employees in this company. Each department has its own information system, database, and data architecture. The database for production, financial, and warehouse are stored in a mainframe from IBM, and the other databases are stored in a workstation running Microsoft NT server. Given that an executive is negotiating with his customers about the price and delivery time of an order, he often needs to know the current status of his company such as order in progress, production capacity and stock level. The system will allow him to just type the question on his PDA screen and get a real-time answer. The executive can also control the operation of his company by giving instructions on the PDA to the system for the system to take initiatives to instruct the relevant staff.

In this case individual data collection agent should be assigned to each department such as DCA-M (marketing), DCA-P (production), DCA-F (financial), DCA-W (warehouse), and DCA-D (distribution). Further, for each DCA, data is supplied by individual data cube. There are three query agents assigned for the respective kinds of enquires such as the capacity (i.e., QA-C), quality (i.e., QA-Q), and stock status (i.e., QA-S). An enjoin agent (EA) is assigned for passing instructions to specified email or IP address. The system when first installed has to retrieve information from different data resources of different functional departments by RTDW. And then it will be continuously updated thereafter.

5.1.2 Operations to Effect Monitoring

The internal operations are described in three phases as follows.

5.1.2.1 Phase I

Upon receiving an enquiry of "capacity: N units of product M at price P delivered to city C by date D" from the end user (marketing department manager on the move), the corresponding QA-C analyzes the received question and organizes four tasks concurrently:

1. communicate with DCA-W to obtain current inventory status of M;
2. communicate with DCA-D for delivery information;
3. inquire DCA-M for price of product M; and
4. monitor the status of requested information from various agents.

5.1.2.2 Phase II

Upon receiving instructions from the QA-C in Phase II, the following three tasks are executed simultaneously in different departments:

1. In the warehouse department, DCA-W queries the inventory database for the inventory status on product M and returns result to QA-C.

2. In the distribution department, DCA-D examines a scenario of scheduling N unit of product M delivered to city C by date D on top of the current delivery schedule, and then returns the result back to QA-C.

3. In the marketing department, DCA-M queries database and obtains current selling price (P') of product M given order quantity N and returns the result back to QA-C.

5.1.2.3 Phase III

Upon receiving all the information from DCA-W, DCA-D, and DCA-M, QA-C exercises its own domain knowledge to evaluate the information and to make recommendations to the user. Based on the overall results, two sets of procedures may be executed by the QA-C:

(i) Case 1—All conditions are met for order acceptance.

(a) QA-C notifies the user that all conditions are met for order acceptance. Then the user will commit the order. Enjoin agent EA will communicate with the DCA-D to trigger a sequence of actions described as follows:

(a-1) DCA-D goes to find out the information for delivery.

(a-2) DCA-D then sends result back to EA to schedule the delivery.

(a-3) EA issues instructions to individual responsible person by inside Email system.

(a-4) EA informs the user the process is successful upon receiving notification from Email server.

(b) QA-C will record the user's decision, which will be used to predict the user's preference in the future.

(ii) Case 2—Not all conditions are met for order acceptance. One of the strengths of MMAP is that agents possess the ability to negotiate among themselves with/without user intervention. Agents will execute a series of negotiation steps if not all conditions are met for order acceptance, as shown in the following example:

(a) If the current price P' of product M is higher than the asking price P, QA-C then sends a request to DCA-F to examine the feasibility of price P* (i.e., the lowest acceptable price) for an order of product M with N units.

(b) QA-C will perform a series of tasks based on results from DCA-F:

(b-1) If P*< P, procedures described in Case 1 will be executed.

(b-2) If P*>=P, QA-C will inform the user about the lowest acceptable price P*. If the customer cannot agree with P*, the executive will make MMAP try to work out another solution.

(c) QA-C will also record the user's decision, which will be used to predict the user's preference in the future.

5.2 Demonstration of the QA prototype

The ASP language combined with Visual Basic is used to implement the system. All agents are coded in VB. Because users may use different mobile devices to access the MMAP, we have developed three types of interfaces suitable for smart phone, PDA and laptop. Users do not need to worry about what device they use. The system is running on a server machine in the backend. Through a wireless network, a mobile device in the front-end is able to automatically detect what device is being used and directly link a suitable format to show the screen for the user. However, to simplify the contents of this thesis, we present all figures only in a PDA form.

To ensure security, we design a login menu that requires users to fill in their username and password. After submitting, the server in the backend needs to check whether it is legal for the user to log in the system. Without permission, the user will be denied access by the system. Besides this mechanism, a wireless access point (WAP) also allows the system administrator to set more than one secure key. Through the double check mechanisms, the MMAP can protect its users from attacks.

For demonstration purpose we present a prototype system that can handle query related to order justification. Given an example of the query "What is the profit for taking the order with price 3.2 and required quantity of 1000?" is inputted into the system, as shown in Figure 5.2.

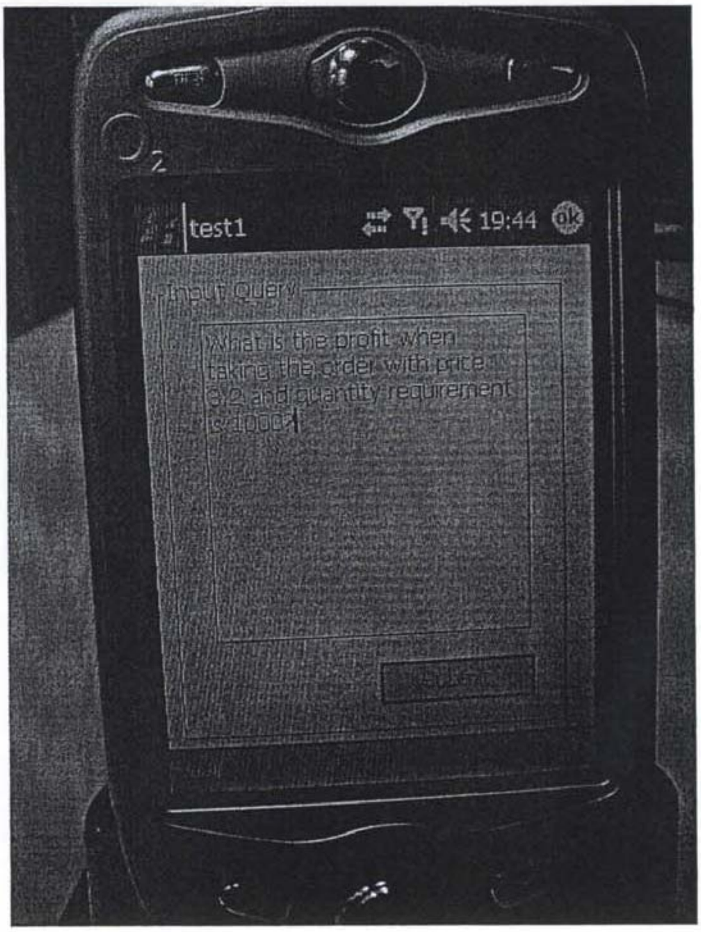

Figure 5.2 Example of Query

After identifying the type of query to be related to order justification, the system will search for the wording of order, profit... etc that are related to the order type query. A form needs to be filled in by the user to confirm the purpose of the query and if the information is available for the query. The following form (shown in Figure 5.3) will be displayed and asked for the input from the user.

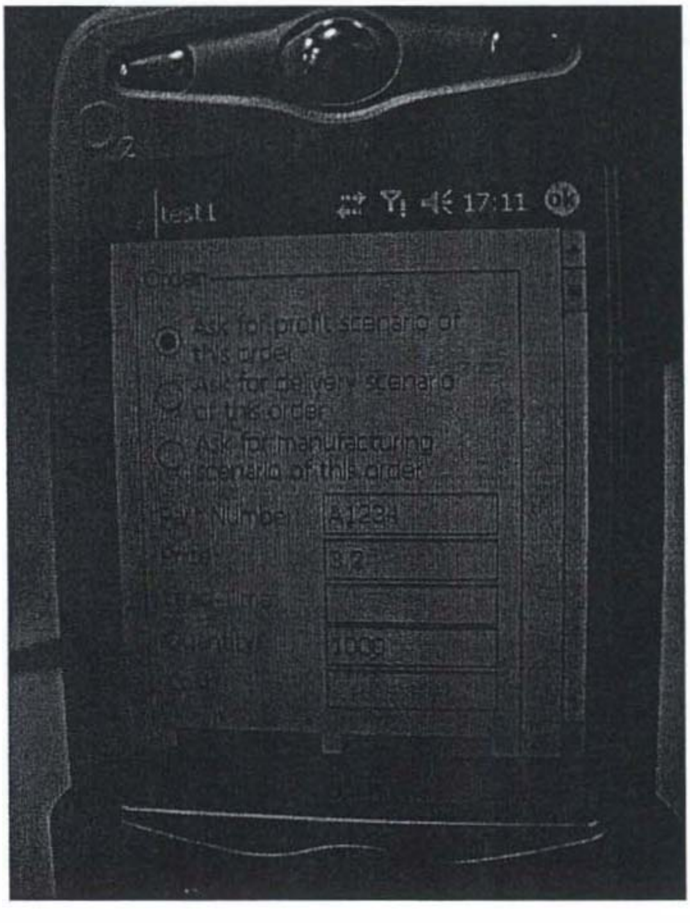

Figure 5.3 Elaborated Form

After filling in this form from the user, we can identify this query as the order type of query asking for profit scenario under different condition. The input conditions are product P/N A1234 with selling price need to be at 3.2 and quantity requirement is 1000. The task is to find out the profit and the lead time under different scenarios. The system will search for most similar case in the sales record and as well as checking whether there is stock available from the warehouse and the outsource availability.

The warehouse database comes back with the available stock of 30000 which satisfy the quantity requirement. The system then check the past sales record with most similar case with such price and quantity requirement and finally the following case is retrieved (the similarity rating is based on the delta between the required price and quantity with each of the cases and those case with the lower average delta would receive higher similarity rating): Jan 2007 with price 2.8 and quantity 1000ea. The cost for this case is 2.3.

Instead of just providing one answer, the system will also provide all available scenarios (although not optimized). In this case the scenario is by outsource and from the warehouse database, it retrieves the outsource cost and delivery.

Based on this information, the system can calculate the profit for this 1st scenario and display the result as shown in Figure 5.4.

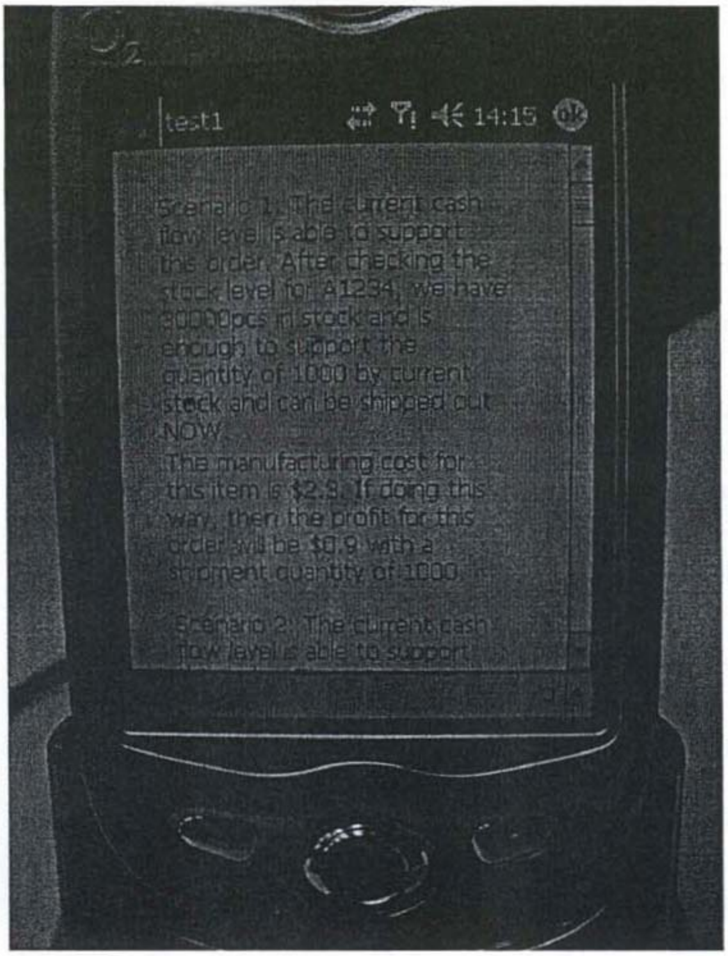

Figure 5.4 Result of Query

Since there may be other factors that may affect the final judgment from the user, so both scenarios are shown to the user which specify the profit and the condition to achieve that profit clearly. The user can base on this information to make the final decision.

System evaluation is the second step in this phase. In order to determine the performance of the proposed algorithms, it is essential to compare it with the conventional resource allocation approach by evaluating the decision-making speed of resource allocation and its effect on order pick-up efficiency. The relevant experimental results are discussed in Chapter 6.

Chapter 6 Conclusions and Future Work

6.1 Summary of Research Work

In recent years, attention has been focused on intelligent systems that have shown great promise in supporting critical managerial decision. In the meantime, emerging mobile technologies such as PDAs, smart phones, and laptops together with wireless networking technologies empower managers on the move to remotely make use of their companies' resources. However, only a small number of the currently used systems are reported to be operating satisfactorily, because many of them are designed to wirelessly access enterprise-wide information systems or retrieve information from a corporate database behind the firewall rather than to provide a complete solution for monitoring and control of a modem enterprise. The aim of this research is to propose an intelligent object paradigm named MMAP to overcome the challenges of demanding executives who seek instant and accurate decision support, and wish to issue real-time instructions to their staff while out of the office or on business trips. The principle and structure of MMAP, featuring the incorporation of RTDW and QCMM, have been developed to deal with the request for information as well as command execution intelligently based on the widely dispersed operation data, for achieving better management performance. The system is driven by intelligent task oriented software agents where the distribution of intelligence is optimized by providing an optimized solution to a nonlinear 0-1 programming model through a scatter search approach.

6.2 The Contributions of the Research

This research has provided a generic object paradigm to guide future development of the core system for the mobile-enterprise. The contributions of this research in improving the management of enterprises are summarized as follows:

(i) MMAP has been developed as a new conceptual framework to provide not only a means for monitoring and controlling an enterprise remotely, but also provide a competitive advantage in giving the executive up to the minute cost and availability data and analysis during negotiations with customers on pricing and delivery time. The innovativeness of the paradigm hinges on a systematic understanding of the various issues involved in the knowledge required for building effective data acquisition strategies and the processing and presentation of these data through the enterprise network to internal and external user applications. It is a new paradigm for automatic capturing and retrieval of information while it also facilitates transmission of instructions across the organization.

(ii) The relevant agent architectures are proposed to carry out the intelligent operation of the system. AI plays an instrumental role in improving the decision making quality and accuracy by distributed knowledge discovery from various operational data. The proposed architecture provides an effective method for codifying, capturing and distributing operation knowledge in an automatic manner. Its focus is to present details through AI techniques for an analysis to be performed on widely-distributed information resources to aid making precise decisions during critical business negotiation.

(iii) A scatter search approach has been developed to optimize the distribution of intelligence in the MMAP so that the best use of available infonnation is exercised to give the most complete answer to a query. The problem to be addressed is not only to assign intelligences to agents, but to appoint units in tasks to intelligences in each agent as well. To maximize the total completion rate of obtained information for a task set, a nonlinear 0-1 programming mathematical model is proposed for formulating the problem. An example and simulation results are conducted, demonstrating that the scatter search approach can be easily and effectively implemented.

6.3 The Limitations of the Research

The limitations of the research are addressed as below.

(i) The proposed system is aimed to provide a mobile solution for remotely managing the enterprise with all operational data available. However, the operational data within an enterprise may sometimes be incomplete. People in different functional areas may not want to share their working practises with other departments, or they are unwilling to input data in real-time due to various private reasons. This situation may affect the completeness of the information/answer given by the system to a user's information request. The proposed system will attempt to give the most complete answer of the available information so far.

(ii) The proposed system requires large amount of data to perform different functions as mentioned in Chapter 3. Therefore, the architecture of the data warehouse should be properly designed for quick and effective queries. The implementation cost of the proposed system becomes high when system users purchase a server with large data storage capacity.

(iii) The user interface can only provide a semi-automatic recognition of the user's request. Natural language translation is not considered.

(iv) The proposed system may adopt RFID technology to collect real time operations data for supporting the decision making activities. However, in order to enable full coverage of the warehouse and shop floor environment, large numbers of RFID devices (RFIID tag, reader and antenna) have to be installed. This implies that high investment cost on RFID technology is needed and hence the proposed system may not be suitable for launching amongst SMEs.

6.4 The Suggestions for Future Work

Some of the characteristics of the proposed object paradigm to be investigated in future research are as follows.

(i) Data integrity problem should be considered in the proposed paradigm to make the most of the available data to provide a solution as accurate as possible.

(ii) An intelligent user interface is required. For example, natural language translation and speech recognition technology may add more capability to solve complicated queries and instructions from the user.

(iii) For the nonlinear 0-1 programming mathematical model, the future study can search for tight lower bounds as these are essential, especially for large instances. Future study may likewise focus on the development of different meta-heuristic procedures to attempt at improving the quality of feasible solutions. Multi-objective optimization for MMAP is also needed to improve the efficiency of the system.

(iv) The applicability of the system on different company structure and core business will be further investigated.

References

Aamodt, A. and E. Plaza (1994). "Case-based reasoning: foundational issues, methodological variations, and system approaches." AI Commun. 7(1): 39-59.

Akerkar, R. and P. Sajja (2010). Knowledge-based systems. Sudbury, Mass., Jones and Bartlett.

Anthes, G. H. (2006). Computer to User: You Figure it Out. Computerworld. 40.

Armstrong, R. (2000). E-analysis for the E-business. Teradata Magazine Online.

Artail, H. A. (2006). "Application of KM measures to the impact of a specialized groupware system on corporate productivity and operations." Information & Management 43(4): 551-564.

Basu, R. (2003). Challenges of Real-Time Data Warehousing. DM Review.

Baum, D. (2006). The Face of Intelligence, ORACLE Magazine.

Becerra-Fernandez, I. and R. Sabherwal (2010). Knowledge management: systems and processes. Armonk, N.Y., M.E. Sharpe.

Ben-Natan, R., O. Sasson, et al. (2004). Mastering IBM WebSphere Portal Server: Expert Guidance to Build and Deploy Portal Applications, John Wiley \& Sons.

Berry, M. J. and G. Linoff (1997). Data Mining Techniques: For Marketing, Sales, and Customer Support, John Wiley \& Sons, Inc.

Bharati, P. and A. Chaudhury (2004). "An empirical investigation of decision-making satisfaction in web-based decision support systems." Decision Support Systems **37**(2): 187-197.

Bhargava, H. K., D. J. Power, et al. (2007). "Progress in Web-based decision support technologies." Decision Support Systems **43**(4): 1083-1095.

Biere, M. (2011). The new era of enterprise business intelligence: using analytics to achieve a global competitive advantage. IBM Press.

Bigus, J. P. and J. Bigus (2001). Constructing Intelligent Agents Using Java, John Wiley \& Sons, Inc.

Bobillo, F., M. Delgado, et al. (2008). "Representation of context-dependant knowledge in ontologies: A model and an application." Expert Systems with Applications **35**(4): 1899-1908.

Bos, J. (2009). "Applying automated deduction to natural language understanding." Journal of Applied Logic **7**(1): 100-112.

Brans, P. (2002). Mobilize Your Enterprise: Achieving Competitive Advantage through Wireless Technology, Prentice Hall Professional Technical Reference.

Bruckner, R. M. and A. M. Tjoa (2002). "Capturing Delays and Valid Times in Data Warehouses\—Towards Timely Consistent Analyses." J. Intell. Inf. Syst. **19**(2): 169-190.

Bui, T. and J. Lee (1999). "An agent-based framework for building decision support systems." Decision Support Systems **25**(3): 225-237.

Burstein, F., C. Holsapple, et al. (2008). Data Quality and Decision Making. Handbook on Decision Support Systems 1, Springer Berlin Heidelberg: 65-82.

Burstein, F. and C. W. Holsapple (2008). Handbook on decision support systems. Berlin, Springer.

Camm, J. D., T. E. Chorman, et al. (1997). "Blending OR/MS, Judgment, and GIS: Restructuring P&G's Supply Chain." INTERFACES **27**(1): 128-142.

Chang, S. E., S. W. Changchien, et al. (2006). "Assessing users' product-specific knowledge for personalization in electronic commerce." Expert Systems with Applications **30**(4): 682-693.

Chen, A. N. K., P. B. Goes, et al. (2006). "Heuristics for selecting robust database structures with dynamic query patterns." European Journal of Operational Research **168**(1): 200-220.

Chen, R.-S., K.-Y. Lu, et al. (2003). "A case study in the design of BTO/CTO shop floor control system." Information & Management **41**(1): 25-37.

Choi, S. P. M., J. Liu, et al. (2001). "A genetic agent-based negotiation system." Computer Networks **37**(2): 195-204.

Chow, H. K. H., K. L. Choy, et al. (2005). "Design of a knowledge-based logistics strategy system." Expert Systems with Applications **29**(2): 272-290.

Choy, K. L., W. B. Lee, et al. (2002). "Development of a case based intelligent customer-supplier relationship management system." Expert Systems with Applications **23**(3): 281-297.

Cimiano, P., P. Haase, et al. (2008). "Towards portable natural language interfaces to knowledge bases - The case of the ORAKEL system." Data & Knowledge Engineering **65**(2): 325-354.

Ciriani, T. A. (1999). Operational research in industry. Hampshire, Macmillan.

Conlon, S. J., J. R. Conlon, et al. (2004). "The economics of natural language interfaces: natural language processing technology as a scarce resource." Decision Support Systems **38**(1): 141-159.

Conlon, S. P., B. J. Reithel, et al. (1994). "A natural language processing based group decision support system." Decision Support Systems **12**(3): 181-188.

Cooper, G. F. and E. Herskovits (1992). "A Bayesian Method for the Induction of Probabilistic Networks from Data." Mach. Learn. 9(4): 309-347.

Coveney, M., D. King, et al. (2003). The Strategy Gap: Leveraging Technology to Execute Winning Strategies, John Wiley \& Sons, Inc.

Cundiff, W. E. (1989). "Database organisation: An algorithmic framework for decision support systems in APL." Decision Support Systems **5**(1): 57-64.

Cus, F., M. Milfelner, et al. (2006). "An intelligent system for monitoring and optimization of ball-end milling process." Journal of Materials Processing Technology **175**(1-3): 90-97.

Datta, A. and H. Thomas (1999). "The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses." Decision Support Systems **27**(3): 289-301.

Davenport, T. H. and J. G. Harris (2005). "Automated Decision Making Comes of Age." MIT Sloan Management Review **46**(4): 83-89.

Davenport, T. H. and L. Prusak (2000). "Working knowledge: how organizations manage what they know." Ubiquity **2000**(August): 2.

De Francesco, N., A. Santone, et al. (2007). "A user-friendly interface to specify temporal properties of concurrent systems." Information Sciences **177**(1): 299-311.

Dhaliwal, J. S. and I. Benbasat (1996). "The Use and Effects of Knowledge-Based System Explanations: Theoretical Foundations and a Framework for Empirical Evaluation." INFORMATION SYSTEMS RESEARCH 7(3): 342-362.

Dolk, D. R. (2000). "Integrated model management in the data warehouse era." European Journal of Operational Research **122**(2): 199-218.

Efendigil, T., O. Semih, et al. (2009). "A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comparative analysis." Expert Systems with Applications **36**(3, Part 2): 6697-6707.

Elisa, B. (1991). Object-Oriented Database Management Systems: Concepts and Issues. M. Lorenzo. **24:** 33-47.

English, L. (1999). Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits, New York: Wiley Computer.

English, L. (2002). The Essentials of Information Quality Management. DM Review. **12**.

English, L. (2005). IQ and Muda. DM Review. **15**: 40-46.

Eriksson, H. (1996). "Expert Systems as Knowledge Servers." IEEE Expert: Intelligent Systems and Their Applications **11**(3): 14-19.

Geoffrey Okogbaa, O., J. Huang, et al. (1992). "Data management system development for on-line data acquisition and process control." Computers & Industrial Engineering **23**(1-4): 341-344.

Geoffrion, A. M. and R. Krishnan (2001). "Prospects for Operations Research in the E-Business Era." INTERFACES **31**(2): 6-36.

Glover, F. (1977). HEURISTICS FOR INTEGER PROGRAMMING USING SURROGATE CONSTRAINTS, Blackwell Publishing Ltd. **8**: 156-166.

Glover, F. (1995). Tabu search fundamentals and uses.

Glover, F. (1998). A Template for Scatter Search and Path Relinking. Selected Papers from the Third European Conference on Artificial Evolution, Springer-Verlag.

Gonzales, M. (2004). The Architecture of Enterprise Data Quality. Intelligent Enterprise. **7**.

Gonzales, M. (2005). Actve Data Warehouses Are Just One Approach for Combining Strategic and Technical Data. DB2 Magazine.

Gorry, G. A. and M. S. Scott-Morton (1971). "A Framework for Management Information Systems." Sloan Management Review 13(1): 55-70.

Gottinger, H. W. and H. P. Weimann (1995). "Intelligent inference systems based on influence diagrams." Decis. Support Syst. 15(1): 27-43.

Guan, S.-U., T. K. Chan, et al. (2005). "Evolutionary intelligent agents for e-commerce: Generic preference detection with feature analysis." Electronic Commerce Research and Applications 4(4): 377-394.

Guo, Z. X., W. K. Wong, et al. (2009). "Intelligent production control decision support system for flexible assembly lines." Expert Systems with Applications 36(3, Part 1): 4268-4277.

Haag, S., M. Cummings, et al. (2007). Management Information Systems for the Information Age, McGraw-Hill Irwin.

Haisten, M. (2005). "The Real-Time Data Warehouse: The Next Stage in Data Warehouse Evolution." from http://www.damaninc.com/company/articles/dwrealtime.htm.

Haq, A., M. Saravanan, et al. (2007). "A scatter search approach for general flowshop scheduling problem." The International Journal of Advanced Manufacturing Technology 31(7): 731-736.

Hayes, K. and S. Kuchinskas (2003). Going Mobile: Building the Real-Time Enterprise with Mobile Applications that Work, CMP Books.

Haykin, S. (1998). Neural Networks: A Comprehensive Foundation (2nd Edition), Prentice Hall.

Heckerman, D. (1996). Bayesian networks for knowledge discovery. Advances in knowledge discovery and data mining, American Association for Artificial Intelligence: 273-305.

Henrik, E. (1996). Expert Systems as Knowledge Servers. **11**: 14-19.

Ho, G. T. S., H. C. W. Lau, et al. (2006). "An intelligent production workflow mining system for continual quality enhancement." The International Journal of Advanced Manufacturing Technology **28**(7): 792-809.

Holland, J. H. (1992). Adaptation in natural and artificial systems, MIT Press.

Holsapple, C. W. (2003). Handbook on Knowledge Management: Knowledge Matters, Vol. 2, Springer-Verlag New York, Inc.

Hopgood, A. A. (2000). Dealing with uncertainty. Intelligent Systems for Engineers and Scientists, Second Edition, CRC Press. **null**.

Huang, G. Q., Y. F. Zhang, et al. (2007). "RFID-based wireless manufacturing for walking-worker assembly islands with fixed-position layouts." Robotics and Computer-Integrated Manufacturing **23**(4): 469-477.

Huber, G. P. (2004). The Necessary Nature of Future Firms: Attributes of Survivors in a Changing World, San Francisco:Sage.

Inmon, W. H. (1992). Building the Data Warehouse, John Wiley \& Sons, Inc.

Inmon, W. H. (1999). Building the Operational Data Store, John Wiley \& Sons, Inc.

Inmon, W. H., R. H. Terdeman, et al. (2001). Data Warehousing for E-Business, John Wiley \& Sons, Inc.

Jennings, N. R. (2000). "On agent-based software engineering." Artif. Intell. **117**(2): 277-296.

Jiang, X. N. and K. L. Yung "A new intelligent system for senior executives to maintain remote control of their company." Expert Systems with Applications **In Press, Corrected Proof.**

Jiao, J., X. You, et al. (2006). "An agent-based framework for collaborative negotiation in the global manufacturing supply chain network." Robotics and Computer-Integrated Manufacturing **22**(3): 239-255.

Kahn, B. K., D. M. Strong, et al. (2002). "Information quality benchmarks: product and service performance." Commun. ACM **45**(4): 184-192.

Kaufmann, E. and A. Bernstein "Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases." Web Semantics: Science, Services and Agents on the World Wide Web **In Press, Corrected Proof.**

Keen, P. G. W. and M. S. Scott-Morton (1978). Decision Support Systems: An Organizational Perspective, MA: Addison-Wesley.

Khan, A. S. and A. Hoffmann (2003). "Building a case-based diet recommendation system without a knowledge engineer." Artificial intelligence in medicine **27**(2): 155-179.

Kim, K.-j. and H. Ahn (2008). "A recommender system using GA K-means clustering in an online shopping market." Expert Systems with Applications **34**(2): 1200-1209.

Kimball, R., M. Ross, et al. (2009). Kimball's Data Warehouse Toolkit Classics: The Data Warehouse Toolkit, 2nd Edition; The Data Warehouse Lifecycle, 2nd Edition; The Data Warehouse ETL Toolk, Wiley Publishing.

Kirby, M. W. (2003). Operational research in war and peace :the British experience from the 1930s to 1970. London, Imperial College Press.

Kirkpatrick, S., C. D. Gelatt, Jr., et al. (1983). "Optimization by Simulated Annealing." Science **220**(4598): 671-680.

Kolanoski, H. (1996). Application of Artificial Neural Networks in Particle Physics. Proceedings of the 1996 International Conference on Artificial Neural Networks, Springer-Verlag.

Kong, S. C. W., H. Li, et al. (2004). "Enabling information sharing between E-commerce systems for construction material procurement." Automation in Construction **13**(2): 261-276.

Kornak, A., J. Teutloff, et al. (2004). Enterprise Guide to Gaining Business Value from Mobile Technologies, John Wiley \& Sons.

Kwon, O. and M. Kim (2004). "MyMessage: case-based reasoning and multicriteria decision making techniques for intelligent context-aware message filtering." Expert Systems with Applications 27(3): 467-480.

Kwon, O. B. and N. Sadeh (2004). "Applying case-based reasoning and multi-agent intelligent system to context-aware comparative shopping." Decision Support Systems 37(2): 199-213.

Laguna, M. and R. Marti (2002). Scatter Search: Methodology and Implementations in C, Kluwer Academic Publishers.

Langseth, J. (2006). "Real-Time Data Warehousing: Challenges and Solutions ", from http://dssresources.com/papers/features/langseth/langseth02082004.html.

Lau, H. C. W., C. W. Y. Wong, et al. (2003). "Design and implementation of an integrated knowledge system." Knowledge-Based Systems 16(2): 69-76.

Lea, B.-R., M. C. Gupta, et al. (2005). "A prototype multi-agent ERP system: an integrated architecture and a conceptual framework." Technovation 25(4): 433-441.

Lee, B.-H., M.-S. Park, et al. (2001). Design and implementation of multi-agent environment for Internet information retrieval service. Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on.

Lee, S.-M. and P. A. Abbott (2003). "Bayesian networks for knowledge discovery in large datasets: basics for nurse researchers." J. of Biomedical Informatics **36**(4/5): 389-399.

Lee, S. M. and S. Hong (2002). "An enterprise-wide knowledge management system infrastructure." Industrial Management & Data Systems **102**(1): 17-25.

Lee, W.-P., C.-H. Liu, et al. (2002). "Intelligent agent-based systems for personalized recommendations in Internet commerce." Expert Systems with Applications **22**(4): 275-284.

Li, S. (2000). "The development of a hybrid intelligent system for developing marketing strategy." Decis. Support Syst. **27**(4): 395-409.

Liang, T.-P. and J.-S. Huang (2000). "A framework for applying intelligent agents to support electronic trading." Decis. Support Syst. **28**(4): 305-317.

Liao, S. H. (2001). "A knowledge-based architecture for implementing military geographical intelligence system on Intranet." Expert Systems with Applications **20**(4): 313-324.

Liao, S. H. (2005). "Expert system methodologies and applications--a decade review from 1995 to 2004." Expert Systems with Applications **28**(1): 93-103.

Lim, J. and K.-H. Lee "Constructing composite web services from natural language requests." Web Semantics: Science, Services and Agents on the World Wide Web **8**(1): 1-13.

Lin, C. and S.-M. Tseng (2005). "Bridging the implementation gaps in the knowledge management system for enhancing corporate performance." Expert Syst. Appl. **29**(1): 163-173.

Lu, K.-Y. and C.-C. Sy (2009). "A real-time decision-making of maintenance using fuzzy agent." Expert Systems with Applications **36**(2, Part 2): 2691-2698.

Madnick, S. and H. Zhu (2006). "Improving data quality through effective use of data semantics." Data Knowl. Eng. **59**(2): 460-475.

Madnick, S. E., R. Y. Wang, et al. (2001). "Improving the Quality of Corporate Household Data: Current Practices and Research Directions." SSRN eLibrary.

Marakas, G. M. (1999). Decision support systems in the twenty-first century, Prentice-Hall, Inc.

Marrs, K. A. and L. G. Robinson (1991). "Object-oriented DBMS requirements." Computer Standards & Interfaces **13**(1-3): 35-39.

Mart, R., M. Laguna, et al. (2006). "Principles of scatter search." European Journal of Operational Research **169**(2): 359-372.

Matsatsinis, N. F., M. Doumpos, et al. (1997). "Knowledge acquisition and representation for expert systems in the field of financial analysis." Expert Systems with Applications **12**(2): 247-262.

Murch, R. and T. Johnson (1999). Intelligent software agents, Prentice Hall PTR.

Negnevitsky, M. (2001). Artificial Intelligence: A Guide to Intelligent Systems, Addison-Wesley Longman Publishing Co., Inc.

Nemati, H. R., D. M. Steiger, et al. (2002). "Knowledge warehouse: an architectural integration of knowledge management, decision support, artificial intelligence and data warehousing." Decision Support Systems **33**(2): 143-161.

Nikolopoulos, K. and V. Assimakopoulos (2003). "The intelligent forecasting infonnation system." Journal of Industrial Management & Data Systems **103**(9): 711-726.

Noh, J. B., K. C. Lee, et al. (2000). "A case-based reasoning approach to cognitive map-driven tacitknowledge management." Expert Systems with Applications **19**(4): 249-259.

Nonaka, I., R. Toyama, et al. (2000). "SECI, Ba and Leadership: a Unified Model of Dynamic Knowledge Creation." Long Range Planning **33**(1): 5-34.

Nwana, H. S. and D. T. Ndumu (1999). A Perspective on Software Agent Research. Knowledge Engineering Review. **2**.

Pacheco, J. A. (2005). "A scatter search approach for the minimum sum-of-squares clustering problem." Comput. Oper. Res. **32**(5): 1325-1335.

Pardoe, D. and P. Stone (2004). "TacTex-03: a supply chain management agent." SIGecom Exch. **4**(3): 19-28.

Park, Y. and S. Kim (2006). "Knowledge management system for fourth generation R&D: KNOWVATION." Technovation **26**(5-6): 595-602.

Pearl, J. (1988). Probabilistic reasoning in intelligent systems: networks of plausible inference, Morgan Kaufmann Publishers Inc.

Peterson, T. (2003). Getting Real About Real Time. ComputerWorld. **37**.

Petit-Roze, C. and E. G. Strugeon (2006). "MAPIS, a multi-agent system for information personalization." Inf. Softw. Technol. **48**(2): 107-120.

Pham, D. T. and D. Karaboga (1998). Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks, Springer-Verlag New York, Inc.

Power, D. J. (2002). Decision Support Systems: Concepts and Resources for Managers, Praeger.

Power, D. J. (2009). Decision Support Basics, Business Expert Press.

Prabhaharan, G., R. Ramesh, et al. (2007). "Concurrent optimization of assembly tolerances for quality with position control using scatter search approach." International Journal of Production Research **45**(21): 4959 - 4988.

Promodel. (1999). "ProModel Saves Dupont $480,000." from http://www.promodel.com/.

Promodel. (2006). "Fiat Case." from www.promodel.com.

Ramamurthy, K., A. Sen, et al. (2008). "An empirical investigation of the key determinants of data warehouse adoption." Decision Support Systems **44**(4): 817-841.

Ras, Z., A. Skowron, et al. (1997). Knowledge discovery from databases with the guidance of a causal network. Foundations of Intelligent Systems, Springer Berlin / Heidelberg. **1325:** 401-410.

Rathnam, S. and M. V. Mannino (1995). "Tools for building the human-computer interface of a decision support system." Decision Support Systems **13**(1): 35-59.

Ray, P. (2000). Cooperative Management of Enterprise Networks, Kluwer Academic Publishers.

Redman, T. (1997). Improve Data Quality for Competitive Advantage. SloanManag Rev. **36**.

Redman, T. C. (2001). Data quality: the field guide, Digital Press.

Respicio, A., M. E. Captivo, et al. (2002). A DSS for Production Planning and Scheduling in the Paper Industry. DSIage2002, Cork, Ireland.

Ribeiro, R. A. and A. M. Moreira (2003). "Fuzzy query interface for a business database." International Journal of Human-Computer Studies **58**(4): 363-391.

Riesbeck, C. K. and R. C. Schank (1989). Inside Case-Based Reasoning, L. Erlbaum Associates Inc.

Rogers, A., E. David, et al. (2007). "The effects of proxy bidding and minimum bid increments within eBay auctions." ACM Trans. Web **1**(2): 9.

Rubenstein-Montano, B., J. Liebowitz, et al. (2001). "A systems thinking framework for knowledge management." Decis. Support Syst. **31**(1): 5-16.

Schuh, G., S. Gottschalk, et al. (2007). "High Resolution Production Management." CIRP Annals - Manufacturing Technology **56**(1): 439-442.

Schwartz, D. G. (2010). Encyclopedia of knowledge management. Information Science Publishing.

Sedighi, A. (2006). "An Indispensable Guide to Knowledge Management Systems." IEEE Software **23**(1).

Shakshuki, E., M. Kamel, et al. (2000). A Multi-Agent System Architecture for Information Gathering. Proceedings of the 11th International Workshop on Database and Expert Systems Applications, IEEE Computer Society.

Sheremetov, L. B., M. Contreras, et al. (2004). "Intelligent multi-agent support for the contingency management system." Expert Systems with Applications **26**(1): 57-71.

Shiu, S. and S. K. Pal (2004). Foundations of Soft Case-Based Reasoning, John Wiley \& Sons.

Simon, H. A. (1977). The New Science of Management Decision, Prentice Hall PTR.

Singh, I., B. Stearns, et al. (2002). Designing enterprise applications with the J2EE platform, Addison-Wesley Longman Publishing Co., Inc.

Song, X., X. Yu, et al. (2009). Model Base Management System Meta Model Design.

Sprague, R. H. and E. D. Carlson (1982). Building Effective Decision Support Systems, Prentice Hall Professional Technical Reference.

Symconidis, A. L., D. D. Kehagias, et al. (2003). "Intelligent policy recommendations on enterprise resource planning by the use of agent technology and data mining techniques." Expert Systems with Applications **25**(4): 589-602.

Szczerbicki, E. and N. T. Nguyen (2010). Smart infonnation and knowledge management. Berlin; London, SpringerLink.

Tang, J., X. Wang, et al. (2008). "Optimization of parts scheduling in multiple cells considering intercell move using scatter search approach." Journal of Intelligent Manufacturing **21**(4): 525-537.

Tapeh, A. G. and M. Rahgozar (2008). "A knowledge-based question answering system for B2C eCommerce." Knowledge-Based Systems **21**(8): 946-950.

Terr, S. (2004). Real-Time Data Warehousing: Hardware and Software. DM Review. **14**.

Turban, E., J. E. Aronson, et al. (2006). Decision Support and Business Intelligence Systems (8th Edition), Prentice-Hall, Inc.

Villa, A. (2002). "Emerging trends in large-scale supply chain management." International Journal of Production Research **40**(15): 3487 - 3498.

Vrba, P., F. Macurek, et al. (2008). "Using radio frequency identification in agent-based control systems for industrial applications." Engineering Applications of Artificial Intelligence **21**(3): 331-342.

Wand, Y. and R. Y. Wang (1996). "Anchoring data quality dimensions in ontological foundations." Commun. ACM **39**(11): 86-95.

Watson, I. (2002). Applying Knowledge Management: Techniques for Building Corporate Memories, Elsevier Science Inc.

Wei, Q. (2002). Agent-Based System for Mobile Commerce.

Wei, Q., T. Sawaragi, et al. (2005). "Bounded optimization of resource allocation among multiple agents using an organizational decision model." Advanced Engineering Informatics 19(1): 67-78.

Wen, W., Y. H. Chen, et al. (2008). "A mobile knowledge management decision support system for automatically conducting an electronic business." Knowledge-Based Systems **21**(7): 540-550.

Wen, W., W. K. Wang, et al. (2005). "A hybrid knowledge-based decision support system for enterprise mergers and acquisitions." Expert Systems with Applications **28**(3): 569-582.

Wess, S., K.-D. Althoff, et al. (1994). Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning. Selected papers from the First European Workshop on Topics in Case-Based Reasoning, Springer-Verlag.

Wilkes, F. M. (1989). Operational research :analysis and applications. London; New York, McGraw-Hill.

Wooldridge, M. (2009). An Introduction to MultiAgent Systems, Wiley Publishing.

Xie, Y., H. Wang, et al. (2005). "A research framework for Web-based open decision support systems." Knowledge-Based Systems **18**(7): 309-319.

Yang, C. C. and F. L. Wang (2007). "An information delivery system with automatic summarization for mobile commerce." Decision Support Systems **43**(1): 46-61.

Youssef, S. M. and R. M. Salem (2007). "Automated barcode recognition for smart identification and inspection automation." Expert Systems with Applications **33**(4): 968-977.

Zeleznikow, J. (1996). "Book review: CASE-BASED REASONING by Janet Kolodner (Morgan Kaufmann Publishers, 1993)." SIGART Bull. **7**(3): 20-22.

Zhang, D. S. and J. L. Zhao (2006). "Knowledge Management in Organizations " Journal of Database Management **17**(1).

Zhen, L., G. Q. Huang, et al. (2010). "An inner-enterprise knowledge recommender system." Expert Systems with Applications **In Press, Corrected Proof.**

Zhen, L. and Z. Jiang (2008). "Knowledge Grid Based Knowledge Supply Model." IEICE - Trans. Inf. Syst. **E91-D**(4): 1082-1090.

Zhen, L., Z. Jiang, et al. (2009). "Knowledge grid-based problem-solving platform." The International Journal of Advanced Manufacturing Technology **42**(11): 1217-1229.

Zopounidis, C., M. Doumpos, et al. (1997). "On the use of knowledge-based decision support systems in financial management: A survey." Decision Support Systems **20**(3): 259-277.