DEVELOPMENT OF A DATABASE MANAGEMENT SYSTEM FOR SMALL AND MEDIUM SIZED ENTERPRISES

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

BY

ÇİĞDEM ŞAFAK

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

APRIL 2005

Approval of the Graduate School of Natural and Applied Sciences

_____

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirement as a thesis for the degree of Master of Science.

_____

Prof. Dr. Kemal İDER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Prof. Dr. S. Engin KILIÇ
Supervisor

Examining Committee Members

Prof. Dr. Metin AKKÖK (METU,ME) _____

Prof. Dr. S. Engin KILIÇ (METU,ME) _____

Prof. Dr. Ömer ANLAĞAN (TÜBİTAK) _____

Prof. Dr. Mustafa GÖKLER (METU,ME) _____

Assoc.Prof. Dr.Tayyar ŞEN (METU,IE) _____

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

Çiğdem Şafak

ABSTRACT

DEVELOPMENT OF A DATABASE MANAGEMENT
SYSTEM FOR SMALL AND MEDIUM SIZED ENTERPRISES

Şafak, Çiğdem
M. Sc., Department of Mechanical Engineering
Supervisor: Prof. Dr. S. Engin Kılıç

April 2005, 132 pages

Databases and database technology have become an essential component of everyday life in modern society. As databases are widely used in every organization with a computer system, control of data resources and management of data are very important. Database Management System (DBMS) is the most significant tool developed to serve multiple users in a database environment consisting of programs that enable users to create and maintain a database. Windows Distributed Internet Applications (DNA) architecture describes a framework of building software technologies together in an integrated web and client-server model of computing.

This thesis focuses on development of a general database management system, for small and medium sized manufacturing enterprises, by using Windows DNA technology. Defining, constructing and manipulating institutional, commercial and operational data of the company is the main frame of the work. And also by integrating "Optimization" and "Agent" system components which were previously developed in Middle East Technical University, Mechanical Engineering Department, Computer Integrated Manufacturing Laboratory (METUCIM) into the SME DBMS, a unified information system is developed.

"Optimization" system was developed in order to calculate optimum cutting conditions for turning and milling operations. "Agent" system was implemented to control and send work orders to the available manufacturing cell in METUCIM. The components of these systems are redesigned to share a unique database together with the newly developed "SME Information System" application program in order to control data redundancy and to provide data sharing and data integrity.

Keywords: Database Management System, Windows DNA, Data Redundancy, Data Sharing, Data Integrity

ÖZ

KÜÇÜK VE ORTA ÖLÇEKLİ İŞLETMELER İÇİN
BİR VERİ TABANI YÖNETİM SİSTEMİNİN GELİŞTİRİLMESİ

Şafak, Çiğdem
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. S. Engin Kılıç

Nisan 2005, 132 sayfa

Veri tabanları ve veri tabanı teknolojisi günümüzün vazgeçilmez öğelerinden biri haline gelmiştir. Veri tabanları bilgisayar sistemine sahip olan kuruluşlarda yaygın olarak kullanıldığından veri kaynaklarının kullanımı ve yönetimi daha da önemli bir hale gelmiştir. Kullanıcılara veri tabanı yaratma ve koruma imkanı veren programlardan oluşan bir veritabanı ortamında birçok kullanıcıya hizmet eden Veri Tabanı Yönetim Sitemi (DBMS) geliştirilmiş en önemli araçlardan biridir. Dağıtık İnternet Uygulamaları (DNA) yapısı ise tümleşik ağ ve alıcı-sunucu modelli bilgisayar uygulamalarında, yazılım teknolojilerini bir arada tutan bir çerçeve oluşturmaktadır.

Bu tez, Windows DNA teknolojisi kullanarak, küçük ve orta ölçekli işletmeler için genel bir veri tabanı yönetim sistemi geliştirilmesini amaçlamaktadır. Çalışmanın ana çerçevesi işletmenin kurumsal, ticari ve üretim prosedürleri ile ilgili verilerinin tanımlanması, yapılandırılması ve yönetilmesidir. Ayrıca, Ortadoğu Teknik Üniversitesi Makine Mühendisliği bölümü Tümleşik İmalat teknolojileri laboratuarında geliştirilmiş olan “Optimization” ve “Agent” sistem bileşenleri yeni geliştirilen sisteme eklenerek bütünleşik bir bilgi sistemi

oluşturulmuştur. “Optimization” sistemi, tornalama ve frezeleme işlemleri için optimum kesme parametreleri hesaplamak için geliştirilmiştir. “Agent” sistemi ise METUCIM’de bulunan üretim birimine iş emirleri göndermek ve bu birimi kontrol etmek üzere tasarlanmıştır. Önceden geliştirilmiş olan bu sistemlerin bileşenleri yeni geliştirilen “KOBİ Bilgi Sistemi” uygulama programıyla birlikte ortak bir veri tabanını kullanabilecek şekilde, veri tekrarını önlemek, veri paylaşımını ve bütünlüğünü sağlamak üzere tekrar tasarlanmıştır.

Anahtar kelimeler: Veri TabanıYönetim Sistemi, Windows DNA, Veri Tekrarı, Veri Paylaşımı, Veri Bütünlüğü.

To My Parents

ACKNOWLEDGMENTS

I would like to express my gratefulness and appreciation to my supervisor Prof. Dr. S. Engin Kılıç for his guidance throughout the completion of this thesis. I would also like to thank Doç. Dr. Tayyar Şen for his suggesstions and comments.

I am also indebted to my colleagues Yusuf Başıbüyük, Burak Sarı, N.Deniz Yücel and Ural Uluer in Integrated Manufacturing Technologies Research Group (IMTRG) for their endless support all through this hard work.

Special thanks go to Koray Tümentümür for his great support and patience througout this study.

I would further like to thank my brother Oktay Şafak for his guidence.

Finally, my greatest thanks go to my parents for their never ending patience.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ADO	ActiveX Data Objects
API	Application Program Interface
ASP	Active Server Page
COM	Component Object Model
DBMS	Database Management System
DFD	Data Flow Diagram
DLL	Dynamic Link Library
DNA	Windows Distributed Internet Applications
ERP	Enterprise Resource Planning
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDEF1X	Integration Definition for Information Modeling
IIS	Internet Information Server
LAN	Local Area Network
MRP	Materials Requirements Planning
MRPII	Manufacturing Resource Planning
MSMQ	Microsoft Message Queue Server
MTS	Microsoft Transaction Server
OLAP	On-Line Transaction Processing
OLE	Object Linking and Embedding
OLTP	On-Line Analytical Processing
OOP	Object Oriented Programming
SME	Small and Medium Sized Enterprises
SQL	Structured Query Language

TCP/IP	Terminal Control Program/Internet Protocol
UDA	Universal Data Access
WfMS	Workflow Management Systems

CHAPTER 1

INTRODUCTION

Information systems are widely used in today's enterprises as their requirements are rapidly changing. Efficient use of knowledge has become one of the major strategies, so the creation of information systems specific to the enterprise demands enables enterprises to manage and disseminate data without redundancy. Several departments in an enterprise use several application programs for different purposes. These programs can use separate databases or a unique database. In this thesis a unified Information System with three application programs using a common database is developed in order to provide data sharing across the enterprise.

1.1 Scope

The scope of this study is to develop a generic database management system for SMEs by using Windows DNA technology. First of all, operational and commercial requirements of a typical SME are analyzed and the entities are identified and defined in order to develop an enterprise data model. Then the data model is constructed that shows entities and associations among those entities. While constructing the data model, two databases "Agent" and "Optimization", previously developed in METUCIM by Cangar [1] and Sarı [2] respectively, are taken into consideration. One of the main objectives of this thesis is the integration of "Agent" and "Optimization" databases into the "SME DBMS" database enabling all of the related application programs work efficiently using a single database. In order to achieve this goal, after analyzing the structure of each

database and designating common entities, a single database is constructed using MS SQL Server 7.0.

Object-oriented modeling approach is used for the development of the methodology assigning the methods for objects to classes written in Visual Basic 6.0. The web site is designed with Visual Inter-Dev 6.0.

The system contains three application programs using a common database; “SME Information System”, ”Agent” and “Optim”. SME Information System is the new developed web-based application program enables users to retrieve or/and update institutional, operational and commercial data of a manufacturing enterprise. All of the manufacturing activities can be easily managed throughout the related web pages by defining new parts, products, process plans and operations. Access to the data about employees and departments of the company is also provided throughout the pages. Pages including the details of purchases, sales and the related companies (suppliers or customers) give the opportunity of following commercial data of the SME.

“Agent” and “Optim” applications previously developed in METUCIM work in the same manner as they work before. The ASP codes and design of a few pages are changed in order to provide conformity with the new database. Also the codes of small programs enabling the graphical output are also modified and recompiled.

The whole system is controlled by the end user (client-tier), which has the necessary access password to the web site. While Internet Information Server (IIS) and Microsoft Transaction Server (MTS) are running on the main server computer (business-tier), SQL server is working on the back up computer (data source-tier).

1.2 Outline

Chapter 2 consists of a survey related to database system concepts, enterprise information systems, modeling techniques and related software technology. Chapter 3 describes SME system model using Windows DNA. Chapter 4 gives detailed information about the development of the SME

Information System. Chapter 5 describes the integration of two other applications called Agent and Optimization into the developed database management system. Chapter 6 includes the test runs. Conclusions and possible future improvements are discussed in Chapter 7. Appendices A, B, C, D and E gives the detailed information on user's manual, keys and ids, sample source code, database tables and database objects of the entire system respectively.

CHAPTER 2

LITERATURE SURVEY

2.1 Database System Concepts

Database systems perform vital functions for all sorts of organizations because of the growing importance of using and managing data efficiently. A database system consists of a software, a database management system (DBMS) and one or several databases. DBMS is a set of programs that enables users to store, manage and access data. In other words database is processed by DBMS, which runs in the main memory and is controlled by the respective operating system [3]. Components of a database system are shown in Figure 2.1.

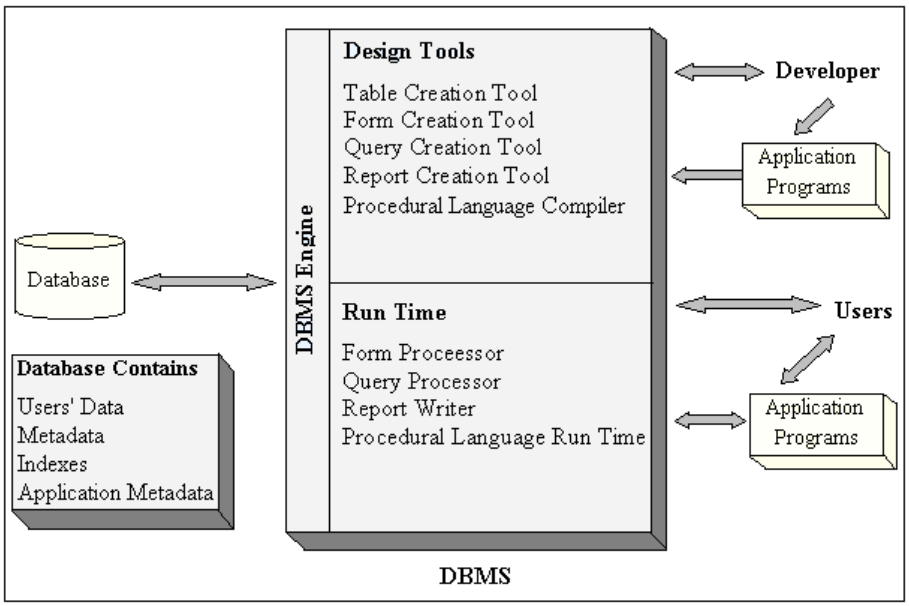


Figure 2.1 Components of Database Systems [4]

A database is a logically coherent collection of data with some inherent meaning and represents some aspects of the real world. A random assortment of data cannot be referred to as a database. Databases draw a sharp distinction between data and information. Data are known facts that can be recorded and that have implicit meaning. Information is data that have been organized and prepared in a form that is suitable for decision-making. Shortly information is the analysis and synthesis of data.

The most fundamental terms used in database approach are "entity", "attribute" and "relationship". An entity is something that can be identified in the users' work environment, something that the users want to track. It may be an object with a physical or conceptual existence. An attribute is a property of an entity. A particular entity will have a value for each of its attributes. The attribute values that describe each entity become a major part of data stored in the database [5].

Figure 2.2 shows two entities, employee and department. Also attributes of each entity are shown. Person_ID, F_Name, L_Name, Birth_Date, Address and Department_No are attributes of Employee entity. Department_No, Department_Name and Department_Location are attributes of Department entity.

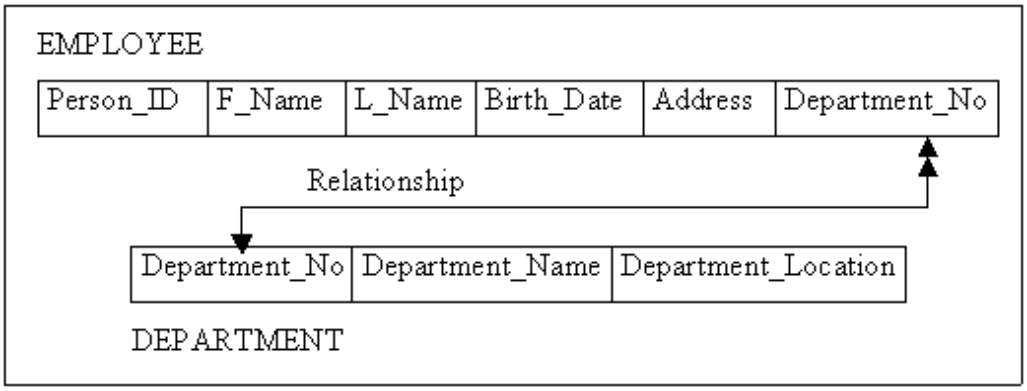


Figure 2.2 Entities, Attributes and One-to-many Relationship

A relationship is an association between entities as shown in Figure 2.2.

There is a relationship between employees and departments. A department is related to all of the employees who work for the company and an employee is related to the department in which he or she works. In this example one department is associated with many employees but each employee is associated with only one department. In this case this is a one-to-many relationship. One-to-many relationship is labeled as 1:M and shown by a double-headed arrow. The other types of relationships used in databases are “One-to-one (1:1) ” and “Many-to-many (M:N) ” relationships. In One-to-one relationship every occurrence of a given entity is related to only one occurrence of another specific entity. An example of one-to-one relationship is shown in Figure 2.3. An employee can manage only one department and that a department has only one manager.

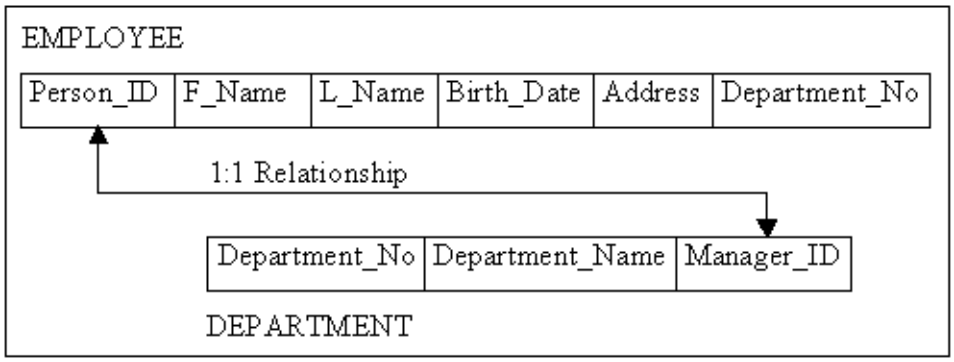


Figure 2.3 One-to-one Relationship

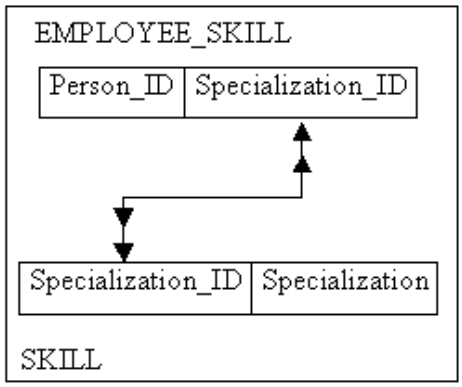


Figure 2.4 Many-to-many Relationship

A Many-to-many relationship describes entities that may have many relationships in both directions. For instance an employee might have more than one job skill and each job skill might be learned by many employees as it is shown in Figure 2.4.

Database Management System is a general-purpose software system designed to manage large bodies of information facilitating the process of defining, constructing and manipulating databases for various applications. Specifying data types, structures and constraints for the data to be stored in the database is called defining a database. Constructing the database is the process of storing data itself on some storage medium that is controlled by the DBMS. Querying to retrieve specific data, updating the database to reflect changes and generating reports from the data are the main concepts of manipulating a database.

The DBMS functions as an interface between the users and the database ensuring that the data is stored persistently over long periods of time, independent of the programs that access it [3].

DBMS can be divided into three subsystems; the design tools subsystem, the run time subsystem and the DBMS engine.

The design tools subsystem has a set of tools to facilitate the design and creation of the database and its applications. Tools for creating tables, forms, queries and reports are components of this system. DBMS products also provide programming languages and interfaces to programming languages. The run time subsystem processes the application components that are developed using the design tools. The last component of DBMS is the DBMS engine which receives requests from the other two components and translates those requests into commands to the operating system to read and write data on physical media. [4]

Database approach has several advantages over traditional file processing in which each user has to create and define files needed for a specific application. In these systems duplication of data is generally inevitable causing wasted storage space and redundant efforts to maintain common data up-to date. In database approach data is maintained in a single storage medium and accessed by various users. The self-describing nature of database systems provides information not

only about database itself but also about the database structure such as the type and format of the data. A complete definition and description of database structure and constraints, called meta-data, is stored in the system catalog. Data abstraction is a consequence of this self-describing nature of database systems allowing program-data independence. DBMS access programs do not require changes when the structure of the data files are changed hence the description of data is not embedded in the access programs. This property is called program-data independence. Support of multiple views of data is another important feature of database systems, which enables different users to view different perspective of database dependent on their requirements. In a multi-user database environment users probably have access to the same data at the same time as well as they can access different portions of database for modification. Concurrency control is crucial for a DBMS so that the results of the updates are correct. The DBMS software is to ensure that concurrent transactions operate correctly when several users are trying to update the same data [5].

Using a DBMS also eliminates unnecessary data redundancy. In database approach each primary fact is generally recorded in only one place in the database [6]. Sometimes it is desirable to include some limited redundancy to improve the performance of queries when it is more efficient to retrieve data from a single file instead of searching and collecting data from several files, but this data duplication is controlled by DBMS so as to prohibit inconsistencies among files. By eliminating data redundancy inconsistencies among data are also reduced [5]. Reducing reduncancy improves the consistency of data while reducing the waste in storage space. DBMS gives the opportunity of data sharing to the users. Sharing data often permits new data processing applications to be developed without having to create new data files. In general, less redundancy and greater sharing lead to less confusion between organizational units and less time spent resolving errors and inconsistencies in reports. The database approach also permits security restrictions. In a DBMS different types of authorizations are accepted in order to regulate which parts of the database various users can access or update.

2.2 Software Technology

In order to build a database system with many components, the software technology should be reviewed for selecting appropriate development programs. Two different approaches to language design can be observed. Approaching from the database perspective, Standard Query Language (SQL) has been the standard for databases for many years. On the other hand, from programming languages perspective there are a number of different object oriented based programming languages used for processing database applications using Internet technology such as C++, Java and Visual Basic. Web-based development kits like Visual Inter-Dev supporting script-languages of JavaScript and VBScript incorporating the database functionality and database management tools like SQL-Server, dBase, Oracle are some of the most widely known major system development tools [3], [4].

In recent years, several new application areas for database systems have emerged. As a consequence, several new data models have been proposed to deal with these new application areas. Object-oriented programming philosophy constitutes the basis for a new approach, the object data model. All of the software development tools mentioned before is utilized in constructing the object data model.

Object-oriented design and object-oriented programming provides a relatively new view of an information system. Object-oriented programming (OOP) is a way of designing and coding programs entailing a new way of thinking about programming structures. OOP views programs as sets of data structures that have both data elements and program instructions instead of viewing programs as sequences of instructions to be processed. The object-oriented paradigm is based on encapsulating data and code related to an object into a single unit. Thus, an OOP object is an encapsulated structure having both attributes and methods. The term encapsulated means that programs external to an object know nothing of its structure and need to know nothing of its structure, that is, it is complete in itself [4]. All interactions between an object and the rest of the system are via messages.

Thus the interface between an object and the rest of the system is defined by a set of allowed messages passing requests among objects and invoking a method [7].

An object-oriented computing platform is the Object Linking and Embedding (OLE) environment represented by Microsoft. OLE is the complete environment of object-based services that supports a common way to design each software piece enabling rich integration between components. OLE provides an infrastructure that supports communication between components and between application programs and components, and incorporate services that are commonly needed by all distributed applications. The concepts that form the idea of an OLE object are collectively called the Component Object Model, or COM. A COM object is one, which supports one or more interfaces as defined by its class. Thus, there could be multiple interfaces to an object. COM objects do not have state; applications obtain a pointer to interfaces that point to the methods that implement them [8]. Also it is called as an ActiveX control which just another term for an "OLE Object" or "COM Object". ActiveX controls are smaller, faster controls that are more suitable for the Internet.

Database processing has always been an important topic in the study of information systems. Internet and the software tools that support the Web have enabled people to use computers in whole new ways. The Internet is a public network of computers that communicate using a communications protocol called Terminal Control Program/Internet Protocol (TCP/IP). Database technology enables Internet applications to be used in a very wide area. And also Internet technology provides a standardized and readily accessible means of publishing database content to users. Internet has given rise to technologies that are used today to publish database applications on the Internet and private intranets. The World Wide Web is one of the Internet's most popular delivery mechanisms using client/server technology in order to access a vast variety of digital information from the Internet. The most common protocol to access these resources is HyperText Transfer Protocol (HTTP) enables the sharing of documents with embedded links to other documents over a TCP/IP network [4]. Internet will be the common platform of almost all of the digital data flowing all through the world

also in the next century making worldwide access possible with minimal cost and complexity.

Client/server computing defines a particular logic and physical environment of computing resources, including hardware and software. The basic idea is that servers connect client stations together and provide services that encourage the sharing of software, hardware, data and processing workload [9]. Client/server environments use a local area network to support network of personal computers, each with its own storage, that are also able to share common devices and software attached to the LAN. The LAN modules of a DBMS add concurrent access controls, possibly extra security features, and query –or translation- queuing management to support concurrent access from multiple users of a shared database. Single or multiple servers act as file or information storage, and interface suppliers responding to client drivers on client's Application Program Interfaces (API's) [10].

Three-tier architecture is a client/server configuration that includes three layers; a client layer and two server layers consisting of a database server and an application server. Figure 2.5 depicts a typical three-tier architecture.

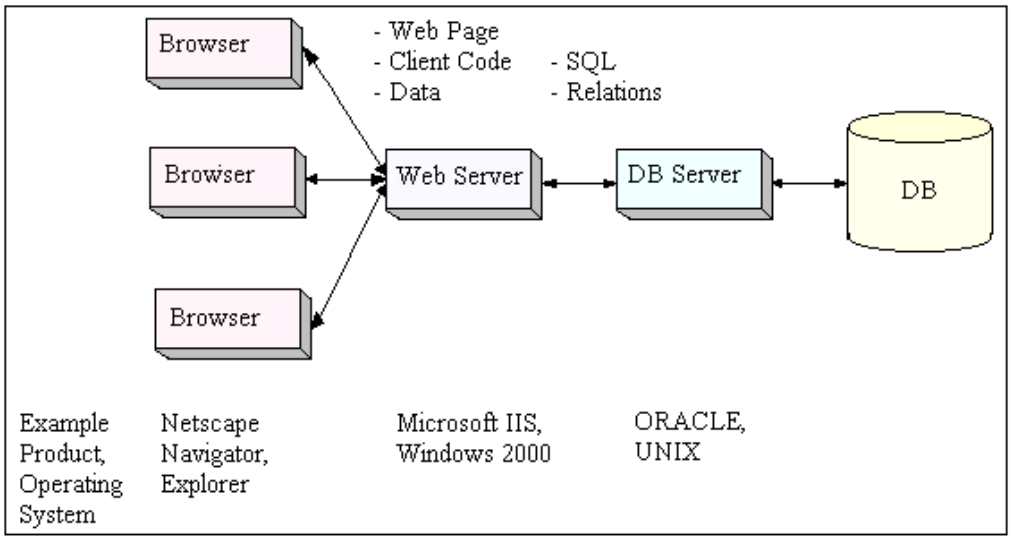


Figure 2.5 Three-tier Architecture [4]

2.3 Review of Enterprise Information Systems:

Information systems have become a crucial requirement for today's enterprises. These systems embody the enabling of the business with capabilities such as global networking, streamlining business processes, sharing information, agility in responding to the market, and intelligent decision making.
Information systems are expected to contribute by:

- Increasing scale efficiencies of firm operations [11], [12]
- Processing business transactions effectively [13]
- Collecting and disseminating timely information for decision making [14]
- Monitoring and recording employee performance effectively [15]
- Maintaining records of business functions within the organization or maintaining communication channels with lower cost [16].

Two important classes of information systems are Enterprise Resource Planning (ERP) Systems and Workflow Management Systems (WfMSs).

2.3.1 Enterprise Resource Planning (ERP) Systems:

An Enterprise Resource Planning (ERP) system is a generic term for an integrated enterprise computing system. It is a customized packaged software based system that handles the majority of an enterprise's information systems requirements. It is a software architecture that facilitates the flow of information among all functions within an enterprise. It sits on a common database and is supported by a single development environment. ERP systems are customized to support an organization's business processes [17]. These systems allow companies to manage multiple operations including manufacturing, human resources, finance and logistics [18].

ERP is defined by Russel & Taylor [19] as an updated Manufacturing Resource Planning System (MRPII) with relational database management,

graphical user interface, and client/server architecture. MRPII is an extension of Materials Requirements Planning (MRP) that plans all the resources necessary for manufacturing; including financial and marketing analysis, feedback loops, and an overall business plan. MRP is a computerized inventory control and production planning system for generating purchase orders and work orders of materials, components, and subassemblies. This brief evolutionary definition of ERP is depicted in Figure 2.6.

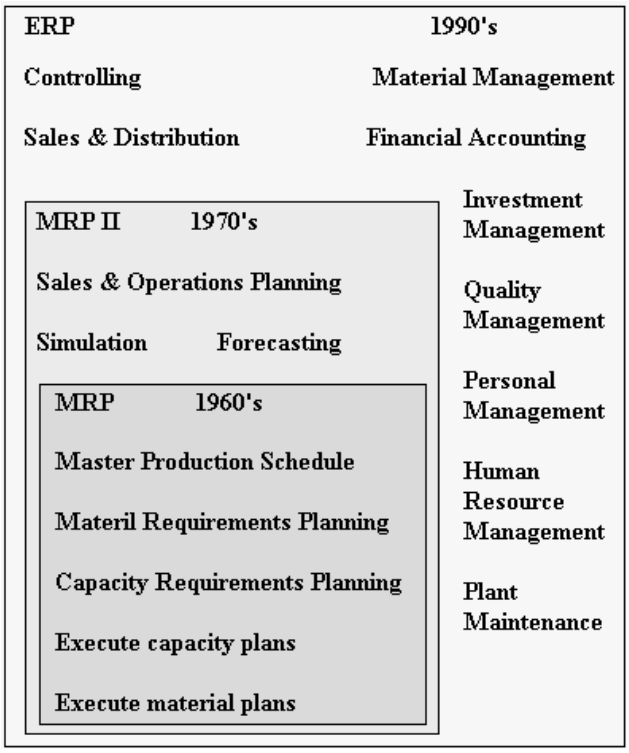


Figure 2.6 ERP Evaluation [19]

As a commercial product, ERP software is offered by a range of vendors like SAP, Baan, J.D. Edwards, PeopleSoft and Oracle [20].

An ERP system differs from an MRPII system in technical requirements such as:

- Graphical User Interface (GUI)

- Relational database
- Client server architecture
- Open system portability

ERP Software provides almost instant access to all the transactional information across the whole organization because the enterprise database enables the unified view and acts as a general resource increasing the visibility of the interdepartmental cooperation and coordination. ERP also eliminates data duplication and inaccuracies. The duplication of data is eliminated while accuracy, reliability, availability and consistency of the data are enhanced. Enhancing the competitiveness of the enterprise as it helps in cost reduction and makes the organization more responsive to customers and markets is another advantage of using ERP [21].

Besides the several advantages of ERP software long implementation times and high costs are serious drawbacks of these systems. Indeed, ERP is often assumed to be a deterministic technology because enterprises are forced to align their work processes with those embedded in the software package [22], [23]. Although the processes embedded in an ERP may be customized through configuration tables, modification of a package's software code to satisfy organizational idiosyncrasies is highly impractical. It is usually necessary for an organization to redefine its business processes to fit the best practices inherent in the software. Thus, ERP is often considered to be a unique kind of technological change, one that is capable of significantly transforming organizations [18].

2.3.2 Workflow Management Systems (WfMSs):

Workflow Management Systems (WfMSs) have been introduced to support the modeling, execution, and monitoring of business processes. Workflow Management Systems allow organizations to define and control the various activities associated with a business process. In addition, many management systems also allow a business the opportunity to measure and analyze the execution of the process so that continuous improvements can be made. Such

improvements may be short-term or long-term. Most workflow systems also integrate with other systems used by the organization: document management systems, databases, e-mail, office automation products, etc. It can also provide a method for organizing documents from diverse sources [23].

Workflow management systems are thus workflow and document management systems that are geared particularly to the specific requirements of public administration.

The core features of WfMSs are [24]:

- Modeling of business process types (workflow schemes)
- Duplication and individualizing of workflow schemes and execution
- Monitoring of ongoing processes and analysis of completed processes.

The advantages of using Workflow Management System within a company are as follows [25]:

- Complete information management within a company
- Increase in work efficiency as a result of better planning, management, controlling and modification of tasks
- Fast and easy introduction of automatic workflow within the company

Under a WfMS, a workflow model is first created to specify organizational business processes, and then workflow instances are created to carry out the actual steps described in the workflow model. During the workflow execution, the workflow instances can access legacy systems, databases, applications, and can interact with users [26].

2.4 Windows Distributed Internet Applications (DNA) Architecture

Windows DNA architecture is Microsoft's framework for building a new generation of n-tier computing solutions. Windows DNA provides a framework for delivering solutions that meet the requirements of corporate computing, the

Internet and intranets, and global electronic commerce, while reducing overall development costs and integrating the Windows world with other platforms both at the front end and the back. Windows DNA Environment is depicted in Figure 2.7.

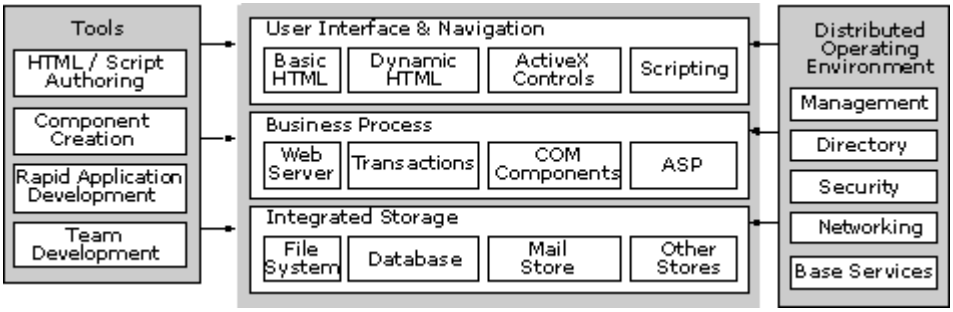


Figure 2.7 Windows DNA Environment [27]

In Windows DNA the applications should be logically separated into partitions, called tiers. Using this methodology increases the scalability, in other words, the software's ability to support a large number of simultaneous users, and also makes the application more manageable and easier to update. Developers must often modify applications to meet new requirements in short order. And how flexible the application will be, will depend on how well the three layers have been separated.

All applications can be broken into three layers [28]:

- Presentation Layer – What the user sees.
- Business Rules Layer - The underlying processing engines and their rules
- Data Layer - The physical data storage layer

The benefit of this architecture is that it gives access to large numbers of users so that they can store and retrieve important data in a consistent and stable manner generally from an application on the client machine.

Windows DNA is based on a programming model called COM (Component Object Model). The COM model has come into widespread use since its introduction by Microsoft and it is an integral part of many Microsoft applications and technologies.

Unlike traditional software development, which required each application to be built from scratch, COM allows developers to create complex applications using a series of small software objects. COM lets developers make portions of their applications using components. This approach speeds up the development process by allowing several teams to work on separate parts at the same time. Developers can also reuse components from one project to the next, and they can easily swap out or update a particular component without affecting other portions of the application. COM also offers the advantage of programming language independence. That means developers can create COM components using the tools and languages, such as Visual Basic, C, C++ and Java [29].

IIS, Internet Information Server, provides the DNA application with the rendering engine needed to publish application content to browsers. By using HTTP to deliver a lightweight presentation layer, the DNA application gains flexibility, quicker implementation and can support and deploy to an unlimited number of clients.

The Business layer (or middleware layer) incorporates the logic for encapsulating the business process. Within this layer it is the responsibility that COM and MTS (Microsoft Transaction Server) interact with other Distributed coordinated services to provide transactions for business "state" information. [28]

MTS ensures that a single transaction is either committed in its entirety or rolled back to its initial state, as though the transaction had never been initiated. The transaction may involve a group of COM objects, anyone of which may abort and so cause the whole transaction to roll back. Furthermore, a single transaction can span objects running on different machines, making MTS ideal for three-tier applications. More than this, MTS provides a complete environment in which to run COM objects. It automatically manages the allocation of threads, connections and other resources, creating instances of your objects as required and disposing of

them when no longer needed. MTS also works with Microsoft Message Queue Server (MSMQ) to support transactions spanning intermittent connections. Messages sent from within a transaction are not put into the receiver's message queue until the transaction commits. If the transaction aborts then any messages that have been received as part of the transaction are returned to the sender's message queue [30].

From within the defined product range Microsoft supplies a production level enterprise database solution, MS SQL Server. SQL Server provides facilities for running databases for any kind of traditional solution including On-Line Transaction Processing (OLTP) and Analytical Processing.(OLAP). SQL Server is tightly integrated into the NT architecture and provides facilities for automatic maintenance and performance optimization.

The primary data source access is provided via ActiveX Data Objects, which are an agreed variant of a COM object, one that supports particular implementation details. ADO is a very flexible data access component that enables anything like relational databases (SQL Server), flat databases (Access), Directory Services (Active Directory), Message systems (Exchange Server) to be a data provider [31]. ADO abstracts the details of the implementation of data access and constructs a consistent data model. Using COM as the delivery mechanism for drivers and consumers means that an agreed supporting infrastructure can be implemented [28].

OLE DB is the underlying technology that ADO and other Data Source technologies use. OLE DB is a specification and implementation written as a low-level COM component that allows more efficient data access [31]. OLE DB enables providers to work against a code base and conceptual framework in order to enable conformance to a data object model that all consumers can access. When all data source providers implement their access in the same manner, through similar standardized interfaces consumers will be able to use a "code template" to access any provider. ADO uses the facilities of OLE DB to provide the unified data access. In fact, ADO is the higher-level data access from Universal Data Access technology group. Within the UDA there are a number of data access

technologies, however, ADO and OLE DB are the primary focus in DNA applications [28].

2.5 IDEF1X Data Modeling Method

The Integration Definition for Information Modeling (IDEF1X) is announced as a Federal Information Processing Standard of United States on November 1985 [32]. IDEF1X is a method for designing relational databases with a syntax designed to support the semantic constructs necessary in developing a conceptual schema [33]

An entity in IDEF1X refers to a collection or set of similar data instances that can be individually distinguished from one another. Individual members of the set are called entity instances. Entities are shown by round-cornered or square-cornered rectangles. Round cornered rectangles represent "dependent" entities those whose unique identifier includes at least one relationship to another entity. "Independent" entities, whose identifiers are not derived from other entities, are shown with square corners. The name of the entity appears outside the box. The box is divided, with identifying attributes (the primary key) above the division and non-identifying attributes below. [34]. Attributes are properties used to describe an entity. Figure 2.8 shows the symbols associated with independent and dependent entities.

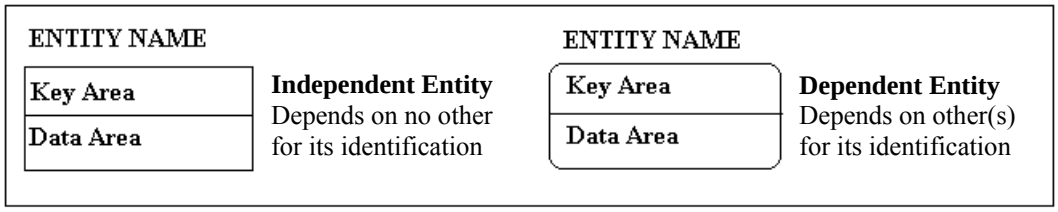


Figure 2.8 IDEF1X Symbols for Independent and Dependent Entities

IDEF1X allows defining relationships between entities. One of the most

important properties of relationships is cardinality ratio. The cardinality ratio specifies the number of relationship instances that an entity can participate in.
Objects may be shared by several objects or they may belong exclusively to one object. It can be specified whether the existence of an entity is depends on its being related to another entity via the relationship type. In order to ensure the retrieval of dependent component objects (child entities) when the parent entity is accessed, it is crucial to define relationships and cardinality ratios. Notation of relationships in IDEF1X is given in Table 2.1.

Table 2.1 Notation of relationships in IDEF1X.

Notation	Relationship Type
A ——— **manages** ——— **B** (Employee) (Department) An Employee can manage only one department	One to one relationship
A ●——— **works for** ——— **B** (Employee) (Department) An employee can work for only one department but a department can have many employees	One to many relationship
A ●——— **works on** ———● **B** (Employee) (Project) An employee can work on more than one project and a project can have several employees.	Many to many relationship

CHAPTER 3

DATABASE MANAGEMENT SYSTEM MODEL

3.1 An Overview

A new model for all of the manufacturing and commercial activities of small and medium sized enterprises using a unique database for three application programs will be discussed in this chapter.

The system software has been implemented in Integrated Manufacturing Technologies Research Group (IMTRG), Middle East Technical University Mechanical Engineering Department Computer Integrated Manufacturing (METUCIM) laboratory. The system mainly consists of a database and three application programs. The first application program "SME Information System" has been newly developed for general requirements of manufacturing companies. The other two application programs "Optim" and "Agent" were developed previously by Sarı [2] and Alataş [35] respectively.

Optimization system software was designed to calculate optimum cutting conditions for single-pass milling and multi-pass turning operations with minimum production cost or minimum production time using different optimization approaches. Different optimization techniques can be used and the results are visualized by graphical methods throughout the web based user interface of the system.

An agent based manufacturing system is an heterarchical, single layered platform with manufacturing system elements, such as parts, and machines, that can communicate between each other, and capable of performing specific tasks in real time. Agents, namely the "intelligent" software modules form a message-based, event-driven scheduling, information processing and execution system.

Agent system software was implemented to control and send work orders to the available manufacturing cell in METUCIM using Agent based communications and DNA technology. The messaging procedure involves a customer-server based negotiation mechanism in which the external input is given by the user (manufacturing engineer) from the locally restricted Internet web site, thus enhancing browsing and monitoring capabilities of online data and status. Any of the scheduling procedures can be easily applied to one or several agents.

A unique database has been constructed for these three applications. Components of the business services of "Agent" and "Optimization" systems have been modified according to the new database and integrated into the SME Database Management System. User-, Business, and Data Services of the system has been mostly written under Visual Basic 6.0. Stateless objects for database search and update has been deployed in Microsoft Transaction Server (MTS). The common database of the system has been constructed using SQL Server 7.0. Internet Information Server (IIS) has been used to grant access to the web sites' ASP and HTML pages, which are designed in Visual InterDev 6.0, a product of Microsoft Visual Studio.

In summary, the developed package includes:

- 71 MTS components for operating optimization algorithms and database search, update, addition and modification,
- 1 SQL Server Database on Internet Information Server with 108 tables and 506 stored procedures,
- Three complete web sites:

1. "SME Information System" newly developed with 36 ASP and 8 HTML pages

2. "Optim" with 34 ASP and 7 HTML pages

3. "Agent" with 22 ASP and 8 HTML pages to control, and manage the system including detailed help and information sources.

The unified information system will enable performing the following tasks:

- Automation of data retrieval,
- Storing and organizing operational data of the SME including process plans, operation steps, types and parameters, machinery and equipment,
- Defining assemblies and their components,
- Storing and defining commercial relations of the enterprise by means of sales and purchases details, sold and purchased items.
- Defining and manipulating all of the organizational data of the enterprise including employees and departments,
- Searching system database according to the selected parameters,
- Defining cutting parameters, making optimization based on maximum production rate or minimum production cost and viewing graphical representations of the optimization results,
- Controlling the manufacturing environment, creating and giving work orders to the system and watching the operations from the web site by web-cam application.

All of the mentioned tasks are summarized in the following Data Flow Diagrams (DFD's):

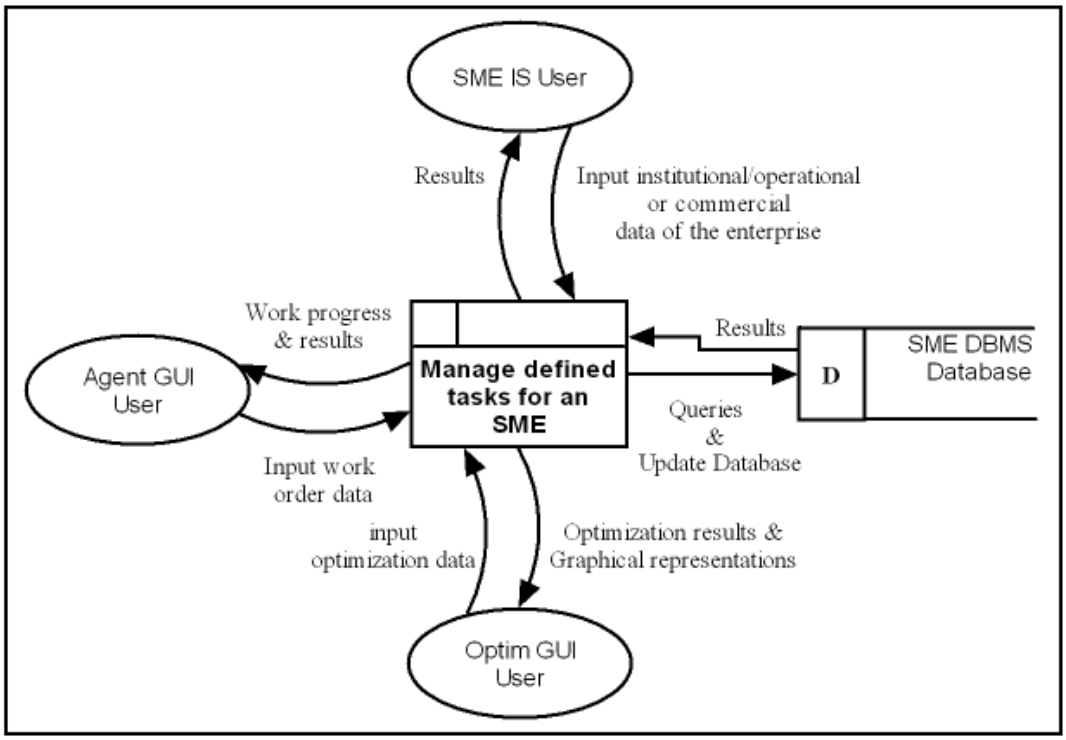


Figure 3.1 Contex DFD of SME DBMS

generalizing the main process of the entire system

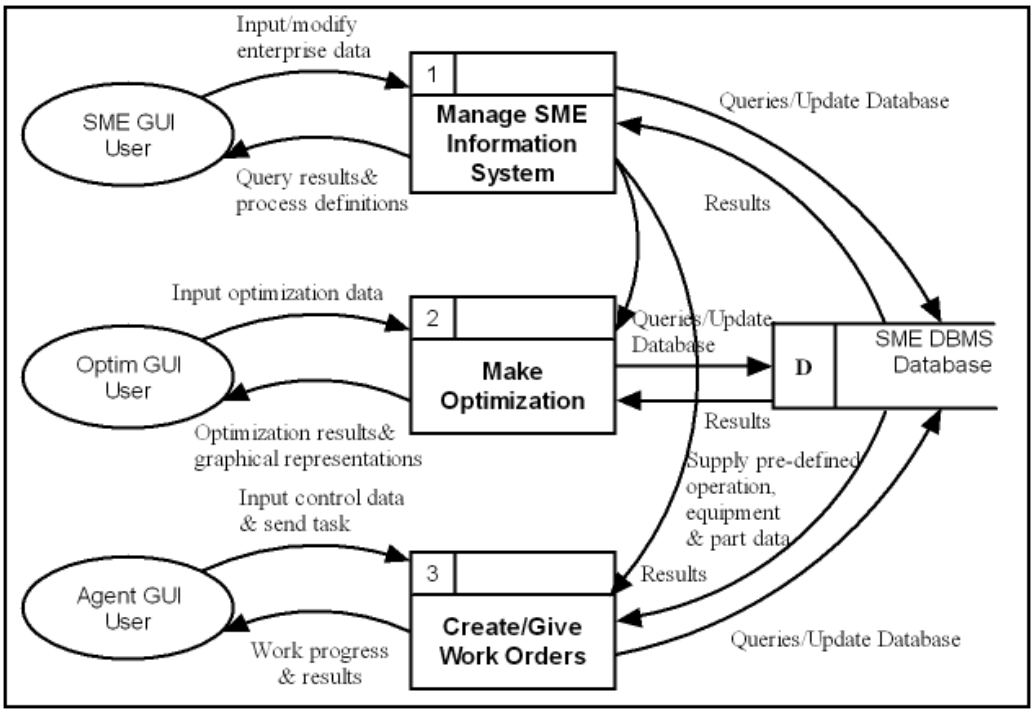


Figure 3.2 Level 1 DFD of SME DBMS

showing the main processes within the entire system

A context diagram is a top level (also known as Level 0) data flow diagram. It only contains one process node (process 0) that generalizes the function of the entire system in relationship to external entities. The context diagram is depicted in Figure 3.1. The main function of the designed system is performing the tasks that are ordered throughout the application programs used within the SME.

The first level DFD contains the main functions of "Agent", "Optim" and "SME Information System" applications. The main process node in the context level is divided into three sub processes, each defining the main process performed throughout the related user interface.

Figure 3.3 and Figure 3.4 show the second and third level Data Flow Diagrams of the system respectively for the processes performed via "SME Information System" application program. The main tasks managed throughout the "Agent" and "Optim" user interfaces are given generally in the first level DFD as it is mentioned before.

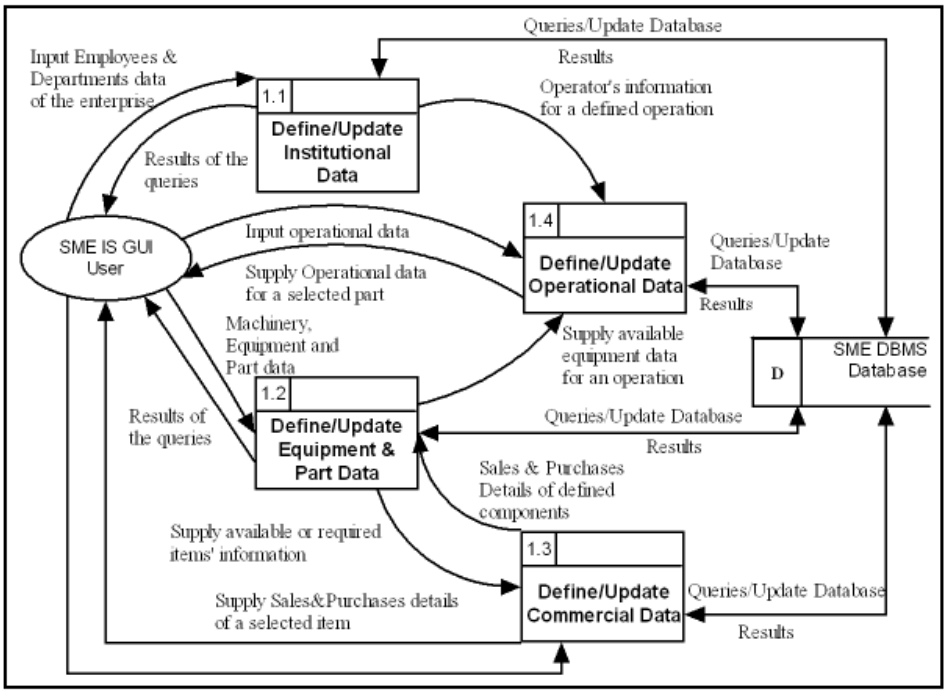


Figure 3.3 Level 2 DFD of SME DBMS (Decomposition of Node1)

In the second level DFD the main process node of the “SME Information System” is broken into four main classes showing the basic processes performed via SME Information System GUI. It is crucial to state each node correctly as these process definitions and the relations among them constitute a basis for the determination of the sub processes that will be performed by the system.

Each process node is decomposed into further processes in the third level DFD showing the detailed inputs and outputs of the performed functions. Although the external entity “SME Information System GUI” and the data storage are not pointed out in Figure 3.4, the relation between user-process and database-process is just as it is shown in other levels.

The main objective of constructing the third level DFD is to identify the functions that will be performed via “SME Information System” application program in detail. An information chain in order to follow processes performed within the enterprise from production to trading is built before the actual design of the system.

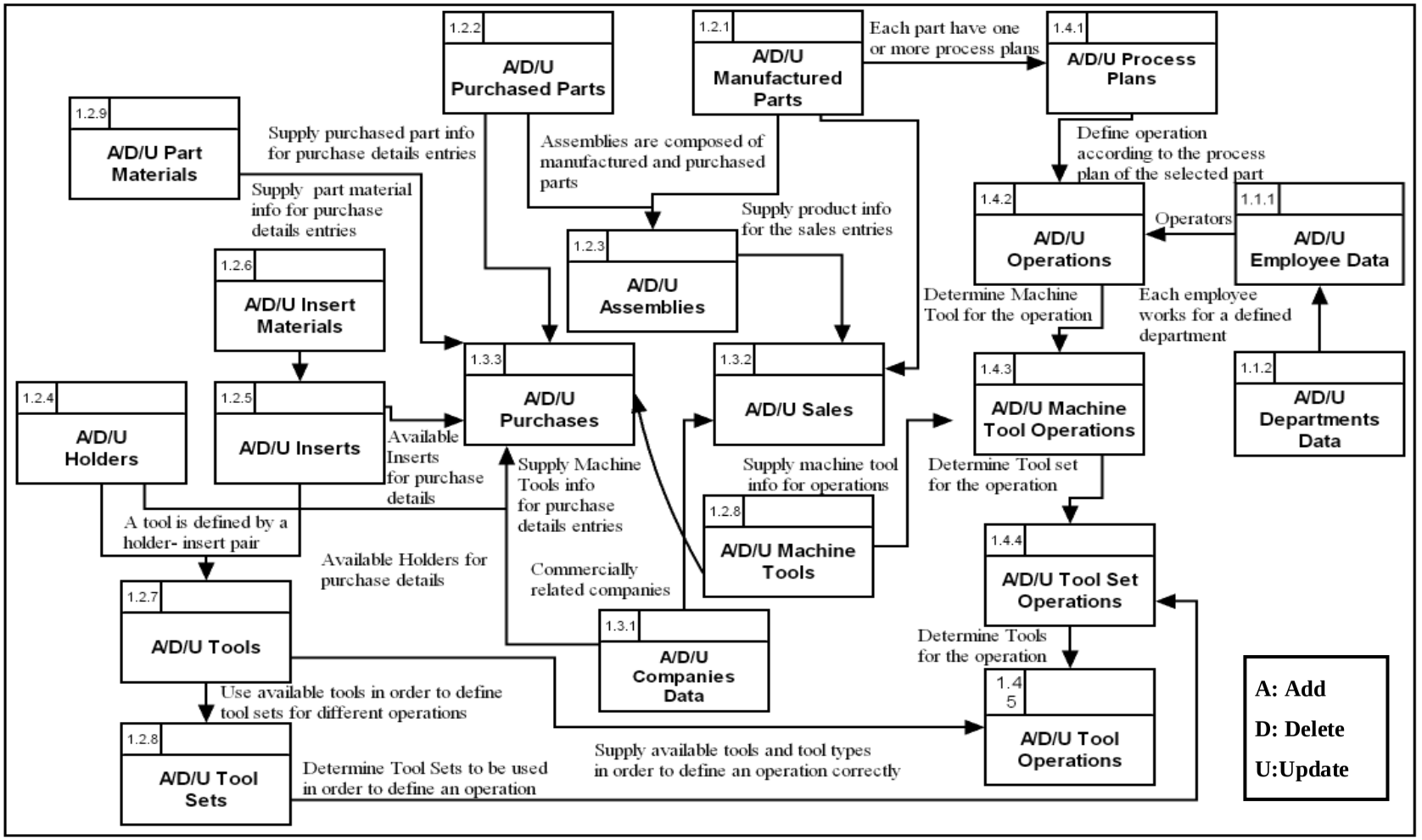


Figure 3.4 Level 3 DFD of SME DBMS showing all of the proceses performed via SME Information System GUI

3.2 DNA Based System Model

Before the programming phase, the logical design of the system is constructed. Although only the components for the "SME Information System" application program are newly created and the existing components used by the "Agent" and "Optim" application programs are added after the required modifications, DNA based system model includes whole system components in order to give a general idea about the structure of the new DBMS.

Agent and Optimization systems components are integrated into the SME DBMS and a unified enterprise information system using a unique database is constructed.

The distributed structure of the system is designed and implemented using the benefits of Windows Distributed Internet Applications (DNA). DNA programming tools provide flexibility in using interfaces with COM, database access with OLE DB, and adaptations to UNIX and mainframes. The three-tier architecture of the whole system is depicted in Figure 3.5.

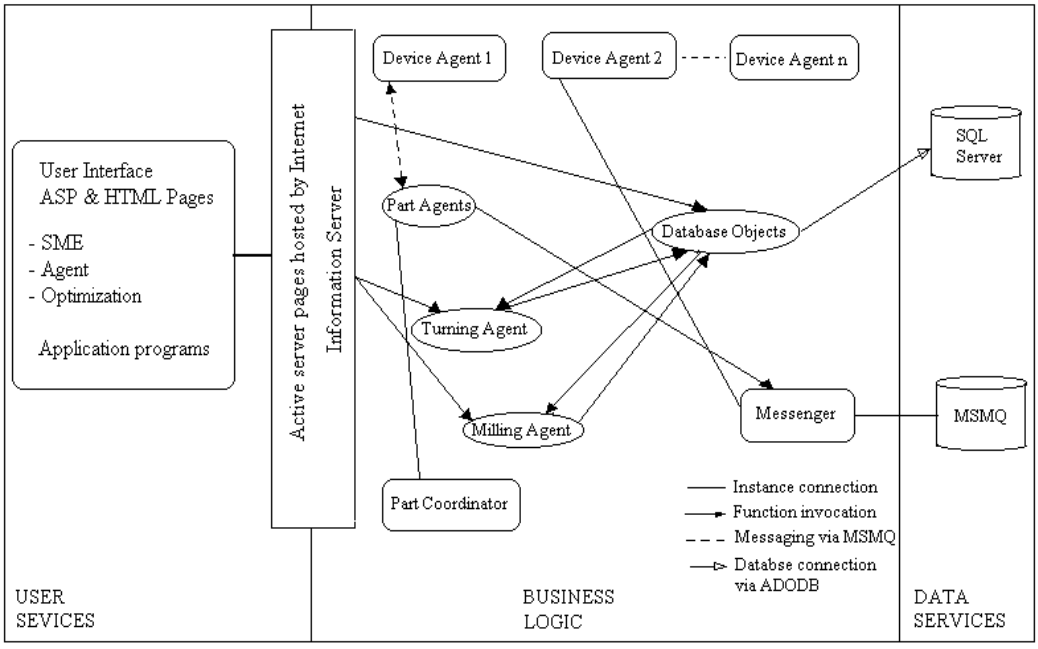


Figure 3.5 DNA Architecture of the System

Three application programs “SME Information System”, “Agent” and “Optim” constitute the presentation tier of the system providing user-friendly interfaces with web-based ASP and HTML pages hosted by Internet Information Server (IIS 4.0). Two of the web sites are modified according to the new developed database and the modified COM DLL components; and a new web site is created for “SME Information System”. Access to all of these applications is provided via Internet, using a username and a password.

The middle tier contains agents and database objects. Agents are previously developed in METUCIM for “Agent” and “Optim” application programs. They are redesigned in order to provide conformity with the new database. The component (business) tier incorporates objects for the optimization of machining operations, machine-part controllers, part coordinator, messaging, database search and updates.

- **Device Agents**, which are independent Win32 executables that run on the PCs, connected to the MSMQ network and the common database. They are the programs that physically control devices in the cell developed by Cangar [1].
- **Part Agents**, which are Transaction Server component DLLs developed by Cangar [1]. They are created and destroyed by the Part Coordinator, which is a Win32 executable running at the backup domain controller hosting to the web site. A part agent is responsible from manufacturing its own. They are dynamic COM components whose life end when their tasks are finished.
- **Milling and Turning Agents** were developed by Sarı [2] and they are Transaction Server component DLLs used by the Web Interface “Optim” to operate optimization algorithms in order to calculate optimum cutting conditions for milling and turning operations. They are created and destroyed by Web Interface. They are dynamic COM components whose life end when their tasks are finished.
- **Database and Messaging objects**, which are objects used by the Web

Interfaces, Device Agents, Part Agents, Turning and Milling Agents to establish connection with the data layer. They are Transaction Server components DLLs. Typically their lifecycle is within the transaction context. They are created, used, and destroyed immediately for a database search, update or a messaging function. Building these “workers” as the Transaction Server components provides an additional safety for database management and messaging.

Table 3.1 DNA Components

Component	**Host**	**Technology**	**Lifecycle**
Part Agents	Primary Backup Controller MTS	DLL	Manufacturing of the part
Device Agent	Local PC's connected to MSMQ	EXE	Existence of the device in the system
DB Objects	Local PC's MTS	DLL	Created and destroyed within transaction context
Messaging Objects	Local PC's MTS	DLL	Created and destroyed within transaction context
Part Coordinator	Primary Domain Controller	EXE	-
Web Interface	Primary Domain Controller IIS	ASP, HTML	-
Data Storage	Primary Backup Controller SQL Server 7.0	Relational Database	-

3.3 Data Model

One of the main goals of this project is to develop a common and flexible database for the applications previously utilized in METUCIM besides the new application program to be developed. In order to provide data sharing and to eliminate update problems, the system has been designed realizing that all of the application programs should use a unique database.

The "Agent" and "Optimization" databases were constructed as separate databases using SQL Server 7.0. The Optimization system was using 39 tables and 227 stored procedures and the Agent system was using 12 tables and 65 stored procedures. As the aim is to construct a common database for three applications it is crucial to identify the systems' data logic. After examining the entities and relationships of these two databases, "SME DBMS" database has been developed in conformity with them using SQL Server 7.0 because of its power and stability. SQL Server is one of the most accepted data tools of DNA family providing management of large amounts of data. SQL Server provides reliability, data integrity, performance, and security and meet most of the expectations while designing a DBMS. Stored procedures, triggers and views make easier to achieve some complex functions.

The requirements of an enterprise working in metal cutting area have been determined. The system has been designed to contain the commercial, operational and institutional data of the company. The data have been stored in a centralized location in a relational database. Relational model is important for two reasons. First, because the construct of a relational model is broad and general, it can be used to express DBMS-independent database designs. Second, the relational model is the basis for almost all DBMS products especially in the area of business and administration applications since it fulfills the demand for managing large amounts of data and providing an intuitive querying and manipulation language. Using IDEF1X modeling technique the data model of the "SME DBMS" has been developed. In this part the logical design of the new constructed "SME DBMS" database is given before combining with "Agent" and "Optimization" databases.

The methodology of constructing a unique database for three application programs is given in Chapter 5. Data model is broken into three parts, each containing institutional, commercial and operational data of the enterprise. The relations among these parts are given in Figure 3.6.

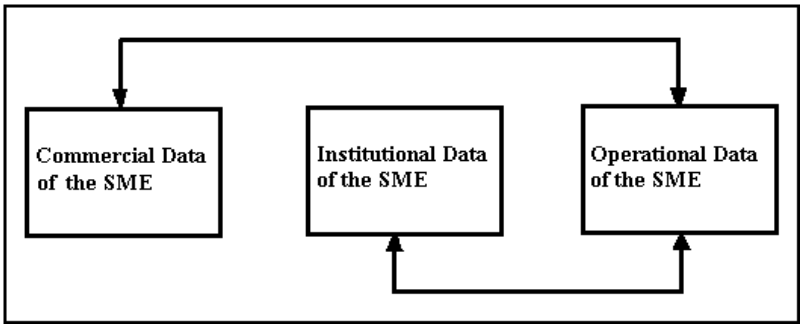


Figure 3.6 Relations among the Data Model parts

Figure 3.7 shows the data tables containing institutional data of the enterprise. IDEF1X data diagrams for operational and commercial data are shown in Figure 3.8 and Figure 3.9 respectively.

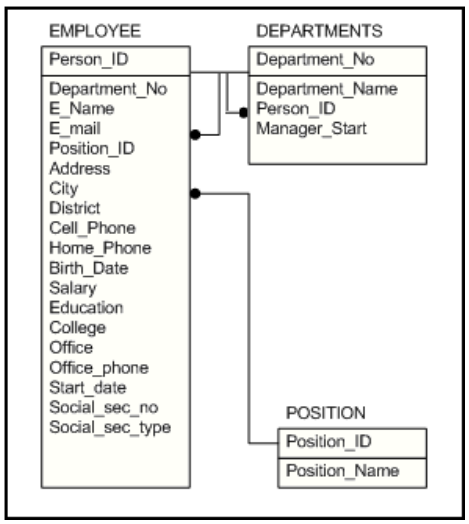


Figure 3.7 IDEF1X Data Model showing the Institutional Data

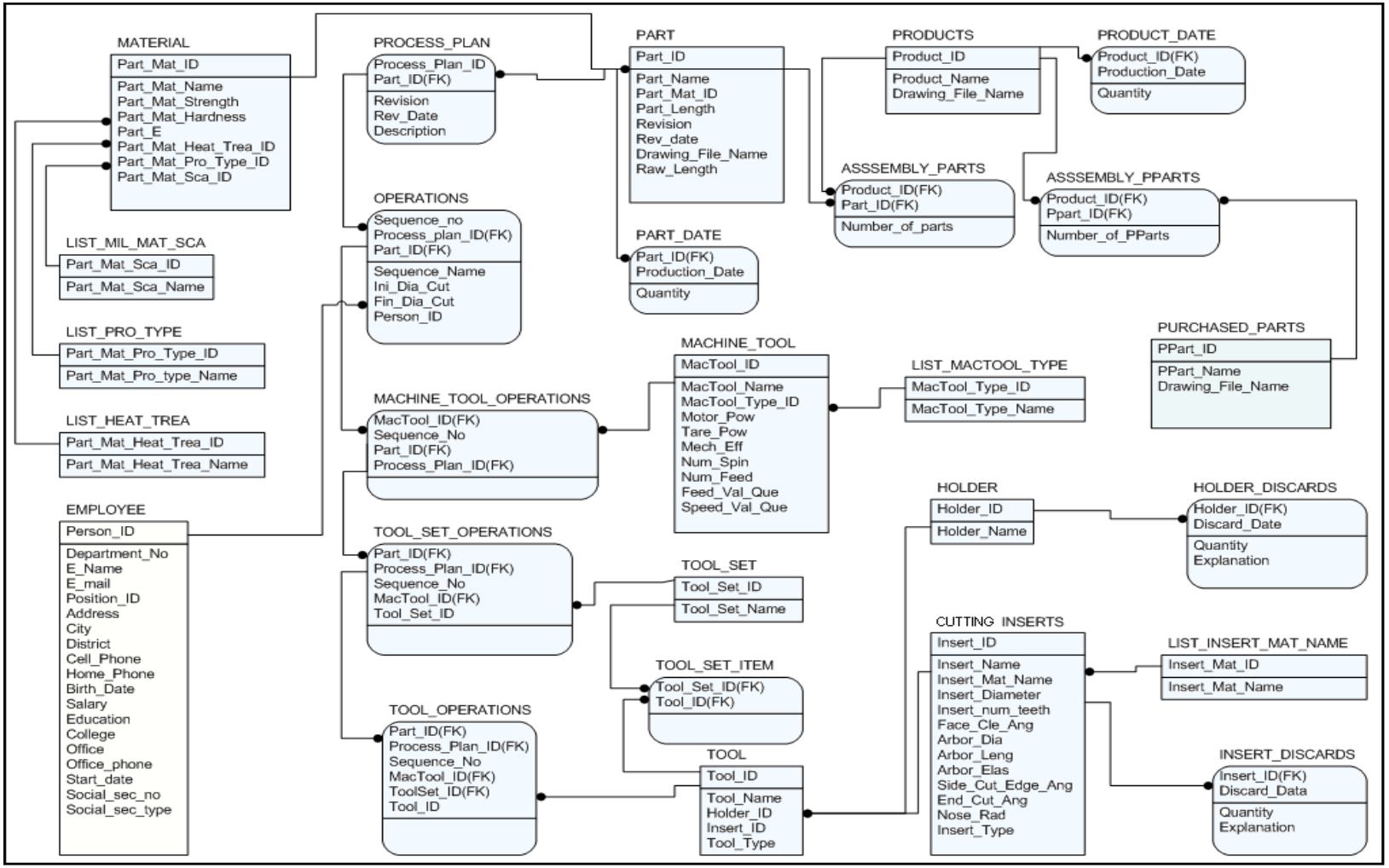


Figure 3.8 IDEF1X Data Model showing the Operational Data

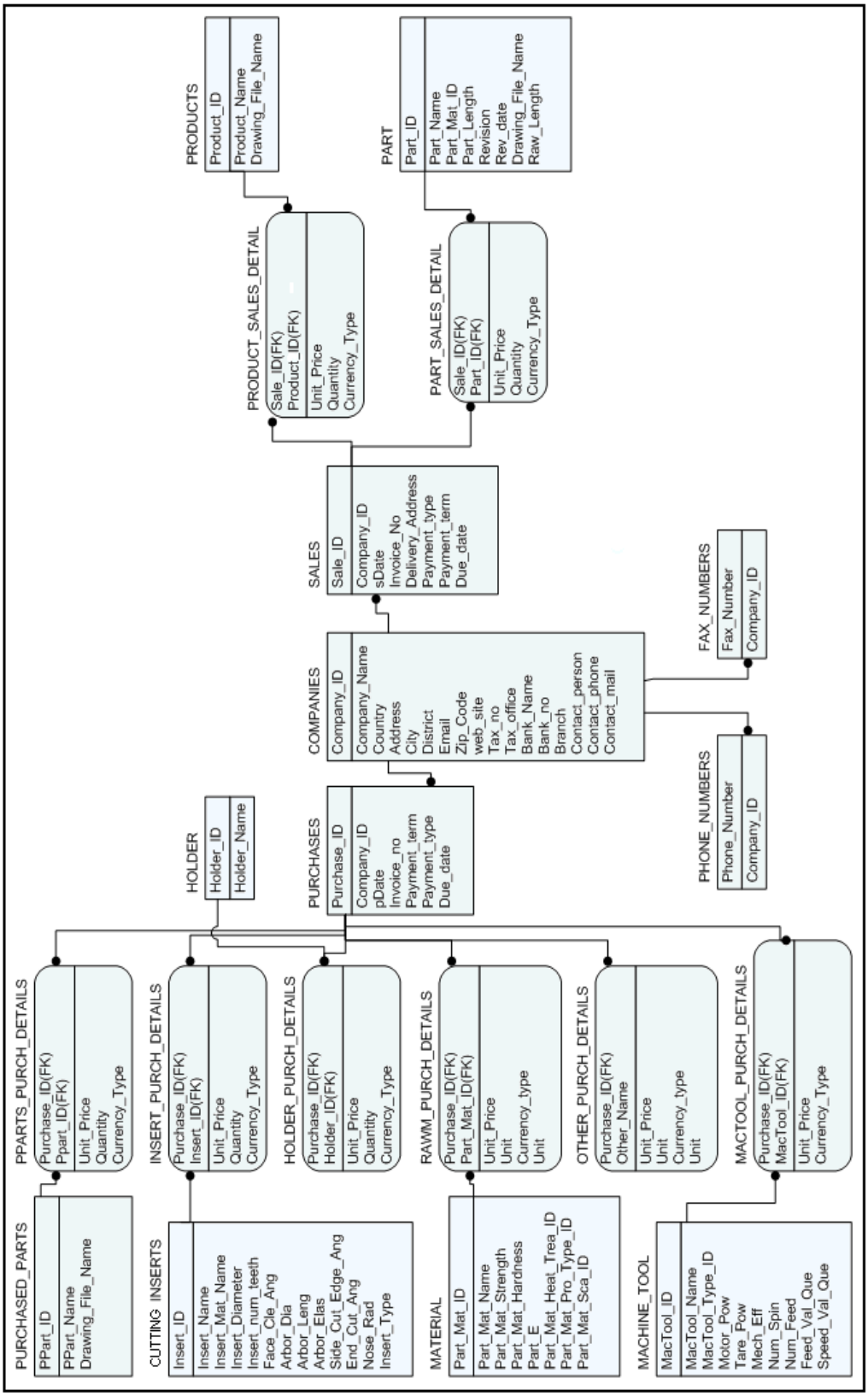


Figure 3.9 IDEF1X Data Model showing the Commercial Data

IDEF1X is a method for designing relational databases with a syntax designed to support the semantic constructs necessary in developing a conceptual schema by defining “entities” and “relationships”. A conceptual schema is a single integrated definition of the enterprise data that is not dependent on any single application and physical storage. IDEF1X is most useful for logical database design after the information requirements are known and the decision to implement a relational database has been made. Hence, the IDEF1X system perspective is focused on the actual data elements in a relational database.

The database model of the SME Information System is constructed using IDEF1X notation as it is shown in the figures. Inheritance and hierarchy between parent and child entities are depicted in the data model diagrams. Due to the relational structure of the database relevant entities are connected to each other via foreign and primary keys providing data integrity.

The system is designed mainly in three parts as it is mentioned before. The first part of the model is designed to contain institutional data of the enterprise. The data model is shown in Figure 3.7. For instance the department, for which an employee works, is represented by the relation between two tables via *Department_No* attribute (foreign key) in *Employees* table and the same attribute as a primary key in *Departments* table.

In the second part of the model operational data is kept as it is shown in Figure 3.8. Although *Employees* and *Purchased_Parts* tables contain institutional and commercial data respectively, they are also included to the data model in order to show the relations among the commercial, operational and institutional data. Blue coloured tables represent operational data tables. Manufactured parts’ data is defined in *Part* table. *Part* table is associated to *Process Plan* table with one-to-many relationship enabling a part having more than one process plans. *Operations, MacTool_Operations, Tool_Set_Operations* and *Tool_Operations* define the operations of a part. They also give a way to visualization with the *Drawing_File_Name* entry of the part and operations. Assemblies are defined in *Products* table and related to *PParts_Product* and *Parts_Product* tables in order to

designate the assembly components. The production dates and the quantities are kept in *Product_Date* and *Part_Date* tables.

The third part contains commercial data. *Purchases* and *Sales* tables consist of attributes containing general sales and purchases information of the enterprise. Although the blue coloured tables contain operational data, they are also included to the diagram in order the show the relationships between operational and commercial data. Green coloured tables represent commercial data tables. *Part_Sales_Details*, and *Product_Sales_Details* tables, containing details of a sale for a specific part or product are connected to *Sales* table via *Sale_ID* attribute and to *Parts/Products* table via *Part_ID / Product_ID* attribute. In the same way the model contains purchase detail tables for different requirements of the enterprise related to the *Purchases* table. *Companies* table have attributes defining the properties of a company commercially related to the SME. From the Figure 3.8 one can inform that one company defined with a *Company_ID* attribute may have one or more *Phone_Numbers* and *Fax_Numbers*.

3.4 System Structure

The objects for data retrieval, deletion and modification are created using Microsoft Visual Basic 6.0 as COM DLLs. The stored procedures created in SQL Server 7.0 are invoked via these objects through the application program. Each object is a class module and has several methods like Add/Delete/Modify/Retrieve. Microsoft Transaction Server (MTS) is designed to host COM DLLs providing an interface to which clients can send requests to perform an action, in response to which they execute the action and send back results to the client. Object caching is a mechanism of MTS, which makes some kind of storage of the functions performed by the object, to rollback if the transaction fails. The objects are created across the network and after performing their functions they are destroyed. The VB code segment to create and destroy an object is given in Appendix C Section C.1.

MTS handles the concurrency, resource pooling, security, context management, and other system-level complexities. The transaction system,

working in cooperation with database servers and other types of resource managers, ensures that concurrent transactions are atomic, consistent, have proper isolation, and that, once committed, the changes are durable.

Transactional safety provided by MTS protects applications from anomalies caused by concurrent updates or system failures. For instance while a user is trying to add a new Product to the database through the application program, if an error occurs or an other user tries to add a new part with the same ID the transaction is rolled back. The *ObjectContex* object is used to control active server pages transactions. An object's associated context object indicates whether the object is executing within a transaction and, if so, the identity of the transaction. The code demonstrating the use of the *ObjectContext* and error handling in a program segment is given in Appendix C Section C.2.

For each table in the database there is a corresponding object in the collection of database objects. The main methods are addition, deletion, and modification of data, search and retrieval in different forms.

CHAPTER 4

DATABASE MANAGEMENT SYSTEM DEVELOPMENT

4.1 An Overview

The logical model of the system has been presented in the previous section with IDEF1X data model and the three-tier structure of the software using DNA technology. This chapter focuses on the work done in building the software. In this chapter the development phases of the system components used by the new developed "SME Information System" user interface is the main area of concern, the methodology of integrating the data and business services components used by the "Agent" and "Optim" application programs into the "SME DBMS" is described in Chapter 5.

Three-tiered model of DNA has been used in building the components of the SME DBMS. Data Services, Business Services and Presentation Services are the components of the system.

Data services keep the records of the system and maintain data integrity using different relationship types between entities. For instance *Products* table is related to *Product_Date* table with a one-to-many relationship and this prevents an entry of nonavailable product to *Product_Date* table.

Business services contain COM DLL database objects enabling different functions such as addition, deletion and modification. For instance adding a new *Employee* or updating the address of an already existing employee.

Presentation services establish the user interface containing the application program codes and enable users to input/display data easily throughout the web pages without the need of database technology knowledge. For example displaying purchase data of a selected *Holder*.

Figure 4.1 shows the three-tiered structure of the system. SME Information System ASP/HTML pages, reachable from anywhere in the world, constitute the presentation services. Database objects are related to both the user interface and the database (data services) of the system. Database objects contains the codes enabling modification of the database that are invoked throughout the user interface.

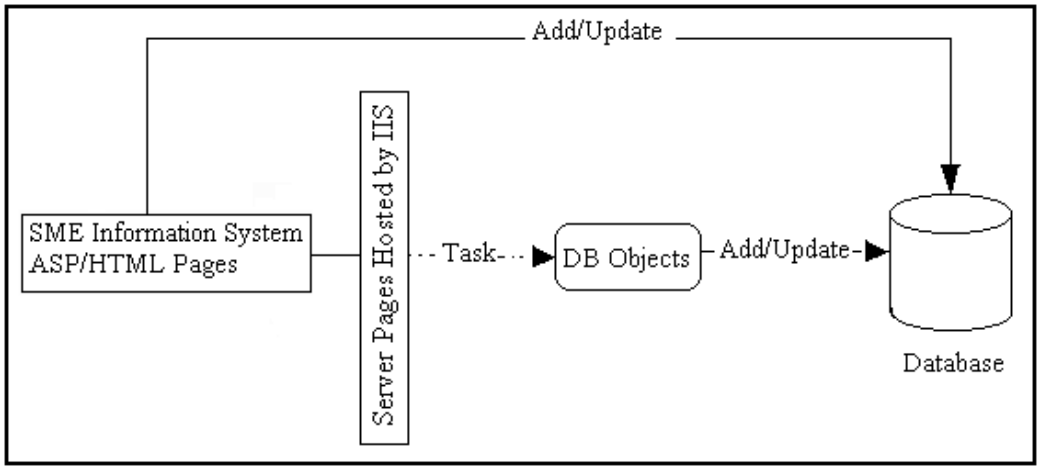


Figure 4.1 Communication between System Components

4.2 Data Services

The database of the "SME DBMS" is constructed using SQL Server 7.0. SQL Server is a product in Windows DNA family, which helps construction of SQL Server relational databases in a systematic and user friendly environment. The Structured Query Language (SQL) is a comprehensive database language that has statements for data definition, query and update. In addition, it has facilities for defining views on the database, for specifying security and authorization, for defining integrity constraints and for specifying transaction controls. It also has rules for embedding SQL statements into a general-purpose programming language such as C or Visual Basic.

The *Create Table* command is used to specify a new table by giving it a name and specifying its attributes and constraints. The definition of a table can be changed by using the *Alter Table* command. SQL has one basic statement for retrieving information from a database, the *Select* statement. The *Insert* command is used to add a single raw to a table and the *Delete* command removes rows from a table. The *Update* command is used to modify attribute values of one or more selected rows.

The logical design of the "SME DBMS" database had been explained in the previous chapter. The physical design of the database is constructed in SQL Server 7.0 as it is mentioned before. The first step is the creation of data tables. The attributes are defined and the primary keys of each table are designated. Each entity is identified by a primary key. For example for the *Employee* table the *Person_ID* attribute is the primary key, that is, it is unique for each employee. Data types and data lengths of each attribute are also defined while creating a table. 49 data tables are created for "SME DBMS" database.

Besides the definition of attributes and keys, constraints and indexes are also created in order to have some other unique data except the primary key. For example in *Tool* table, shown in Figure 4.2, *Tool_ID* is the primary key that identifies a tool and a tool consists of a holder and an insert. In order to prevent data duplication *Insert_ID* and *Holder _ID* attributes together are defined as a unique constraint. If this constraint had not been created, an already existing holder-insert set would have been defined as a new tool only with a different *Tool_ID*.

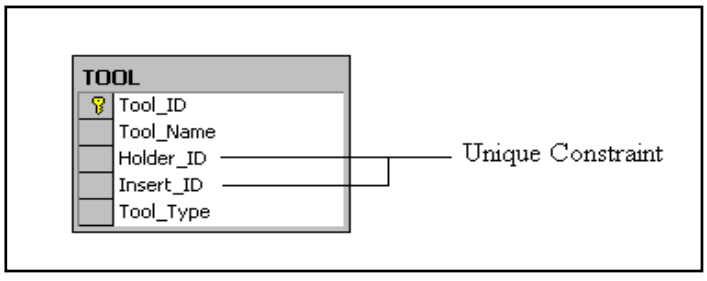


Figure 4.2 Tool Table

After the creation of data tables, the database diagram is constructed. Database diagrams enable defining relationships among data tables, in other words among entities. The database diagram of the “SME DBMS” constructed in SQL Server 7.0 is shown in Figure 4.3. A list of database tables and attributes are given in Appendix D.

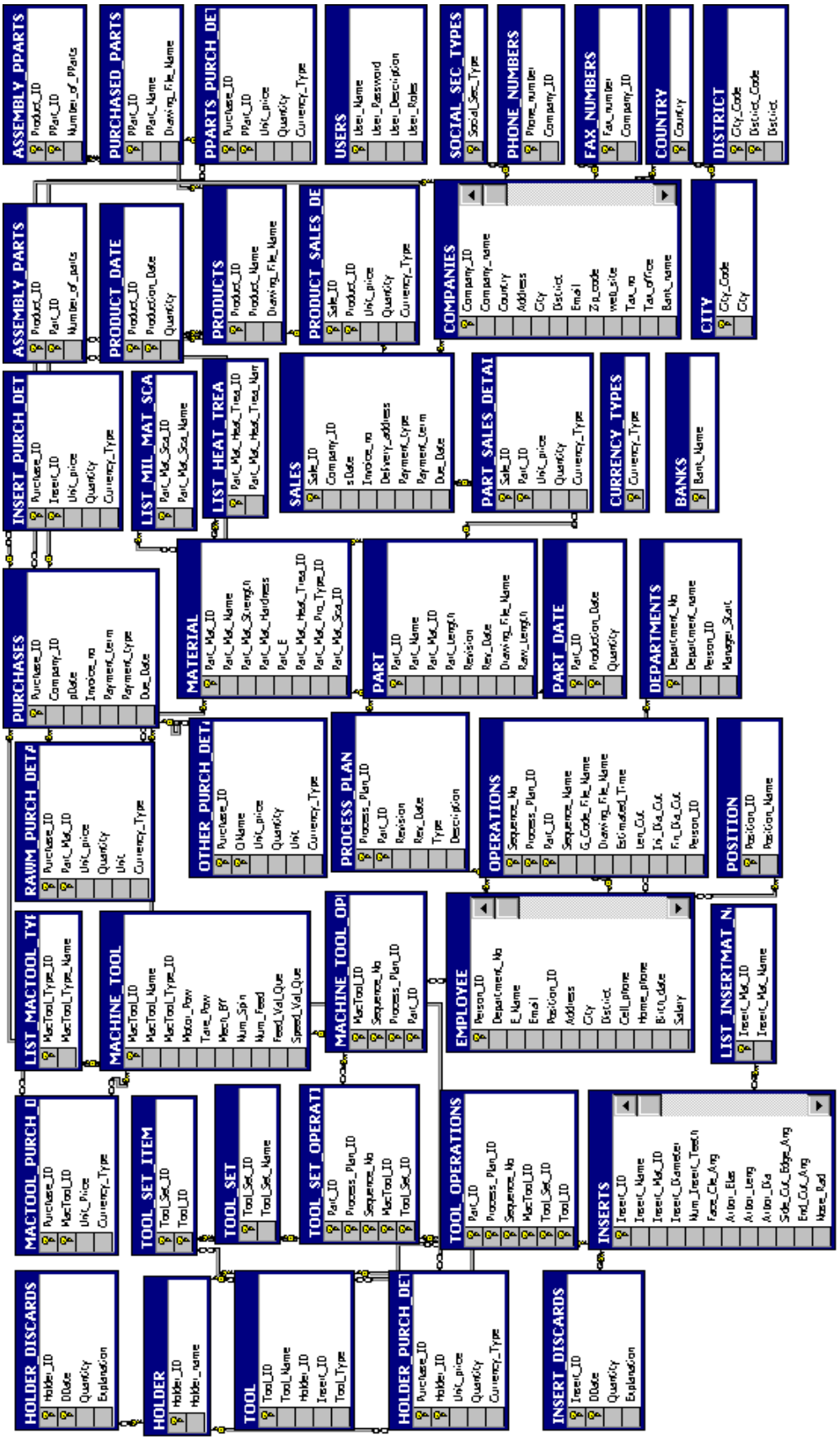


Figure 4.3 The Database Diagram of the "SME DBMS"

In the physical database design phase of the system another aspect is the creation of stored procedures. Parameters may be passed to a query to perform searches according to the user inputs using stored procedures. In order to add, delete, modify or retrieve data stored procedures are invoked through the user interface. The COM DLL database objects contain code pieces for calling a stored procedure. Examples of the created stored procedures are for data retrieval and addition are given in Appendix C Section C.10.

In order to achieve the task the database object containing the part addition method is called throughout the application program and the stored procedure is invoked. A new part defined with a *Part_ID*, with its attribute values entered from the application program, is added to the part table.

SQL Server gives the opportunity of creating views combining the selected tables due to the selected parameters. *Inner Join* operation is executed after selecting the tables and the required attributes from each table. Design view of *VIEWProductSale* is given in Figure 4.4.

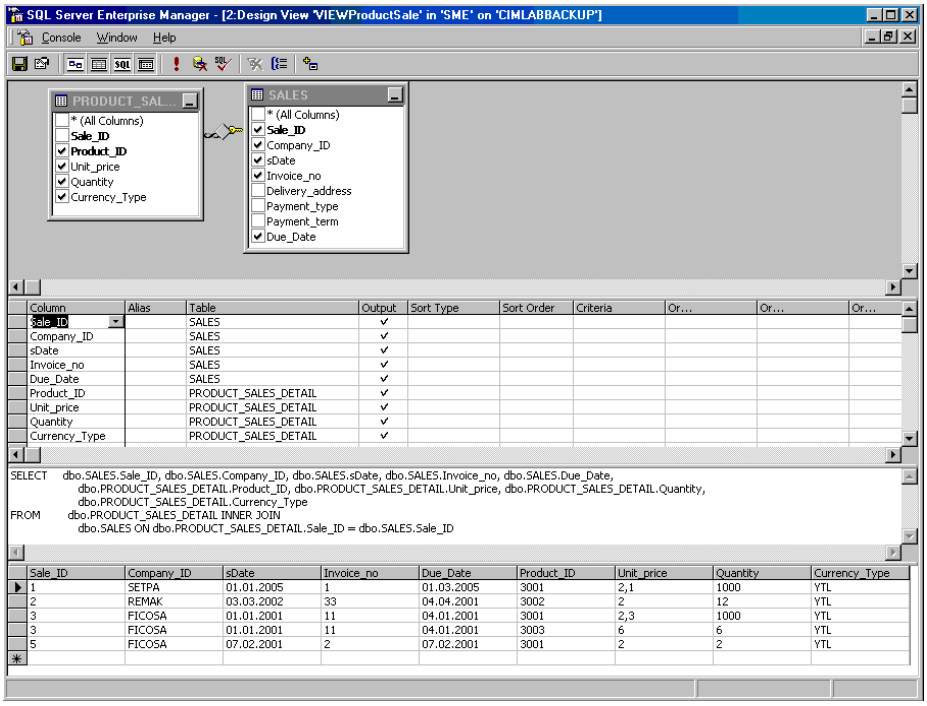
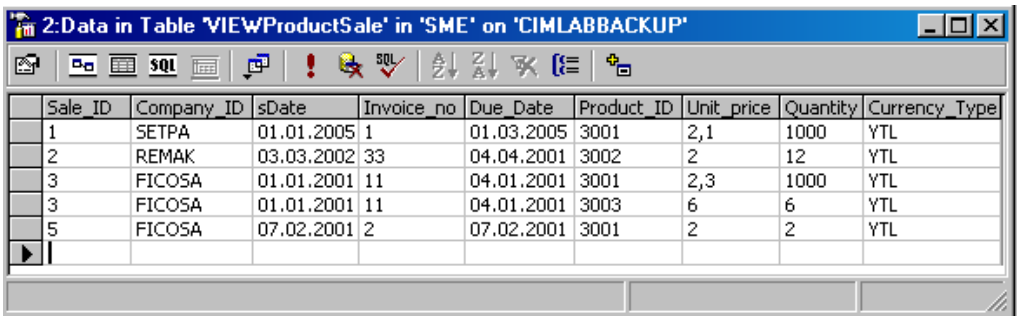


Figure 4.4 Design view of "VIEWProductSale"

In this example *Sales* and *Product_Sales Details tables are joined* in order to retrieve the data having common Sale_IDs, thus the related attribute values of the selected Sale is taken from two tables. For example *Product_Sales_Detail* table does not contain the company data to which the product is sold. Creating the view enables joining them in a single table. After selecting the tables and attributes SQL Server 7.0 automatically generates the *Create View* statement. The generated inner join statement code can be found in Appendix C Section C.7.

The result of the query seen at the bottom of the design view screen after running the function is given in Figure 4.5.



Sale_ID	Company_ID	sDate	Invoice_no	Due_Date	Product_ID	Unit_price	Quantity	Currency_Type
1	SETPA	01.01.2005	1	01.03.2005	3001	2,1	1000	YTL
2	REMAK	03.03.2002	33	04.04.2001	3002	2	12	YTL
3	FICOSA	01.01.2001	11	04.01.2001	3001	2,3	1000	YTL
3	FICOSA	01.01.2001	11	04.01.2001	3003	6	6	YTL
5	FICOSA	07.02.2001	2	07.02.2001	3001	2	2	YTL

Figure 4.5 VIEWProductSale

4.3 Business Services

The core component of the system as its name reveals is the “SME.cls”. SME database objects of are created using Visual Basic 6.0. For each table in the database there is a corresponding database object constructed using class modules. Database objects are used to add, delete, modify, and retrieve data from the SQL Server database. The objects are created and destroyed almost immediately once they complete their action.

An object has methods, properties or collections. An object’s methods determine the functions like Add, Delete, Modify or Retrieve. The methods of the database object *SME.MacTool_Purch_Details* are given in Figure 4.6.

An object's properties can be set to specify the object. For instance the properties of *SME.MacTool_Purch_Details* object are *Purchase_ID,* MacTool*_ID*, *Unit_price* and *Currency_Type* same as the attributes of the data table *MacTool_Purch_Details.*

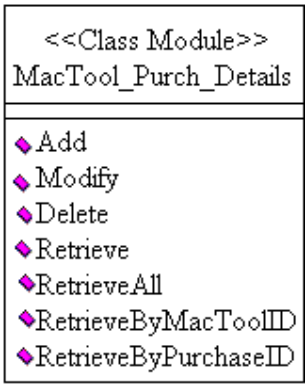


Figure 4.6 Methods of the *SME.MacTool_Purch_Details* Object

The collection of an object contains different sets of keys and value pairs related to the object. For instance *Purchase_ID* and *MacTool_ID* attributes forms the key set of *SME.MacTool_Purch_Details* object. Each value of a key set has corresponding *Unit_Price* and *Currency_Type* values. In Database objects, key attributes are defined by using *ByVal* before the attribute name while non-key attributes are defined by *Optional ByVal* as it is shown in the following example defining the add method of the object:

Public Function Add (ByVal Purchase_ID As Long, ByVal MacTool_ID As Long, Optional ByVal Unit_price As Currency, Optional ByVal Currency_Type As String) As String

Database use generally the stored procedures in order to query data. The related stored procedure in SQL Server is invoked using database objects. A sample code for this method is given in Appendix C Section C.8.

Database objects are Microsoft Transactions Server components. An overview of the main MTS screen is shown in Figure 4.7.

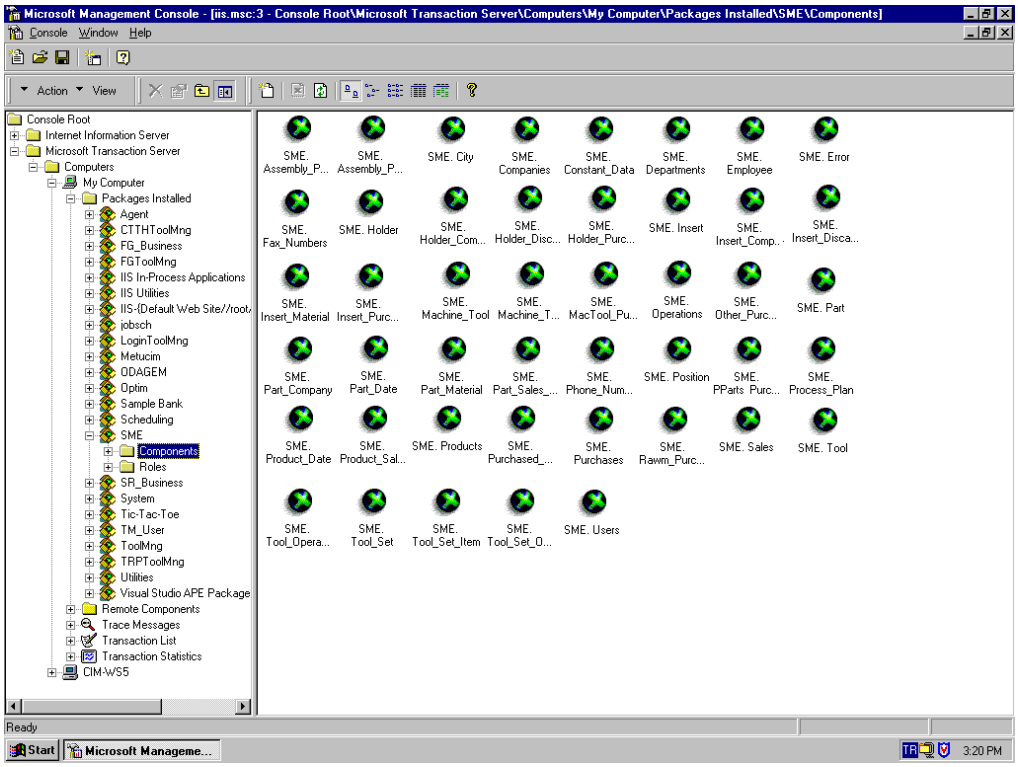


Figure 4.7 MTS Components Screen

4.4 Presentation Services

The web-based interface of the system is implemented using the ASP programming model. VB-Script is used throughout the pages. Visual tools such as buttons, figures, combo and textboxes are applied for a user-friendly interface.

The “SME Information System” is the web based user interface of the system enabling access to the commercial, operational and institutional data of the enterprise. Also Ms. Word reports on employees, supplier and customer companies can be generated. Several reports can be added to the system using Visual Interdev and preparing Ms. Word templates.

Based on the defined user roles, different functions can be performed throughout the web pages. The users are prompted to indicate a valid username and password to enter the web site. By that, the identity of the current person is detected and stored, also unattended entries to the system is prohibited. The main entrance to the system is from “Login “ page.

User rights are based on:

1. **Administrator:** Represents users that can fully administer all information contained in the database.
2. **Engineer**: Represents users that can add, remove, delete and update the enterprise data but cannot change the system structure.
3. **User:** Represents users that cannot change any database entry but can browse all information.

After login to the system the “Root” window that contains links to the accessible web pages appears.

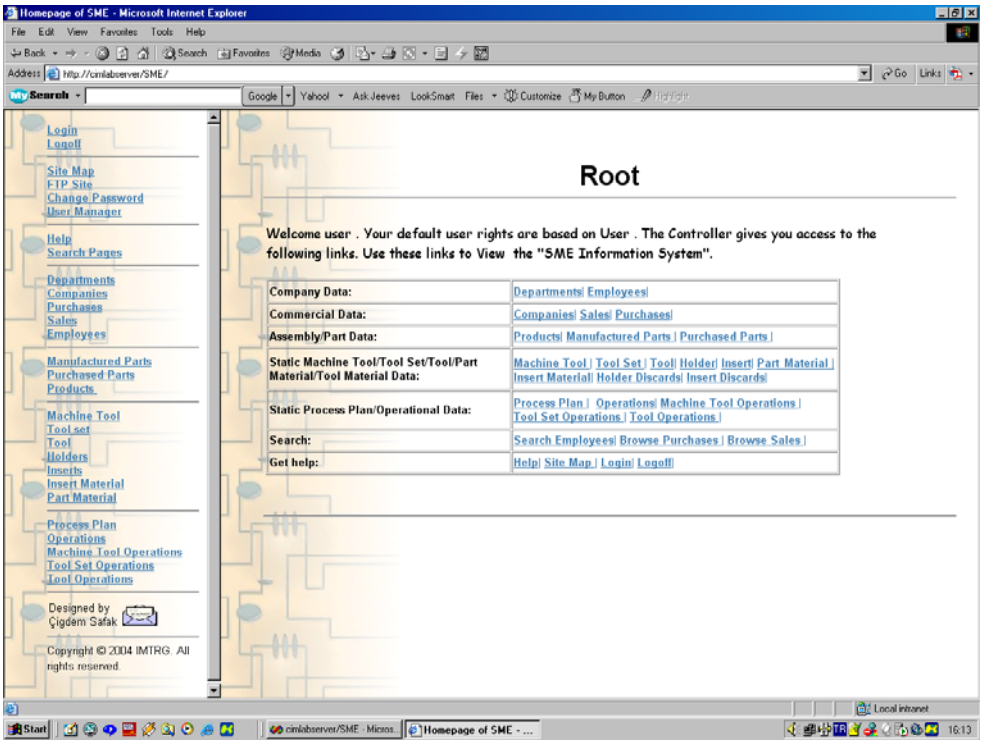


Figure 4.8 Root Window

As it is shown in Figure 4.8 the pages can be classified into seven categories.

1. *Company Data* category includes the pages containing institutional structure data of the company. *Departments* and *Employees* web pages constitute this category. *Employees* page visualize the detailed information about a selected employee according to his/her position and name.

2. *Commercial Data* category indicates the pages providing access to the Companies page containing information of the companies that have commercial relationships with the SME, and also to the Purchases and Sales screens. Commercial activity of the SME can be viewed or updated easily throughout these pages. Besides the general information about a selected sale, the details about the sold parts and products can be viewed throughout the sales page. Purchases page contain commercial details about parts, raw materials, holders, inserts, machine tools or any other item that are purchased by the enterprise.

3. *Assembly/Part Data* shows available products, manufactured and purchased parts with drawings. The production dates and quantities of assemblies and manufactured parts can also be viewed or updated. Products page also shows the components that constitute an assembly.

4. *Static Machine Tool/Tool Set/Tool/Part Material/Tool Material Data* shows available machine tools, tool sets, tools, holders, inserts, part and tool materials in "SME Information System".

5. *Static Process Plan/Operational Data* indicates, process plans for available parts, process plan operations with NC codes, connections between machine tools and these operations, between tool sets and indicated machine tools, tool

sets and related tool. Also persecution of insert and holder discards is possible throughout the related pages.

6. *Help:* Links to help, login, logoff and web site map screens. Help screen providing access to sitemap, thesis documents and step by step instructions is shown in Figure 4.9.

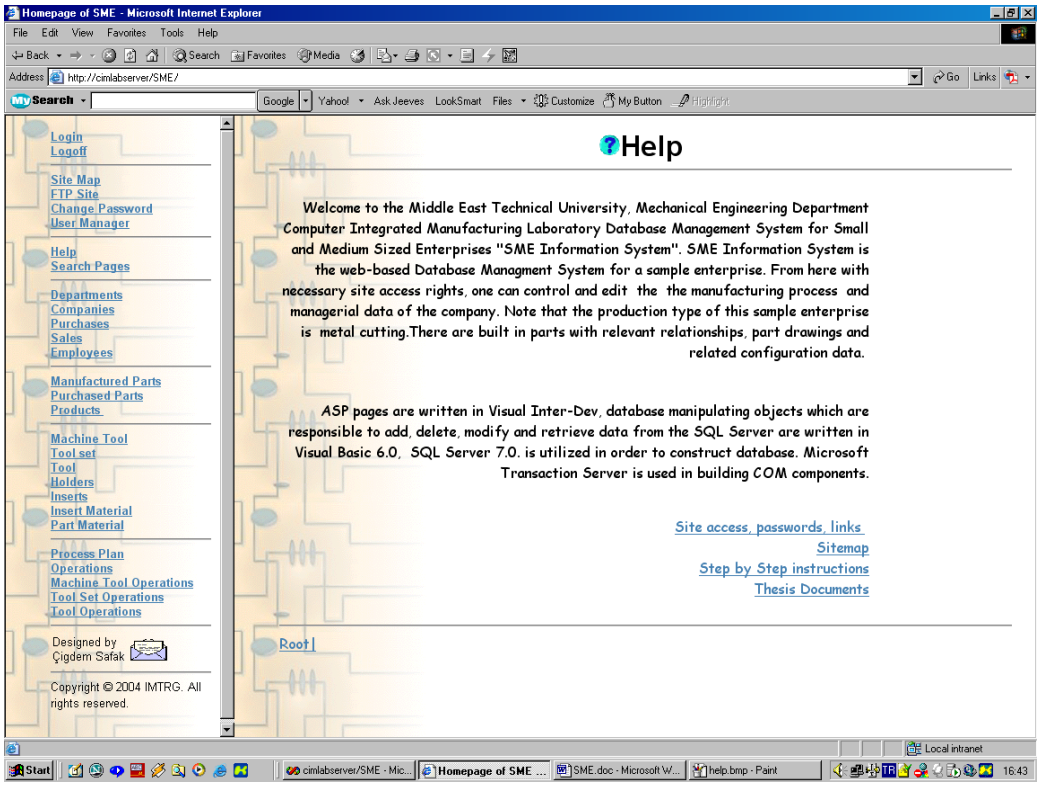


Figure 4.9 Help screen

7. *Search:* Links to Search Employee, Browse Purchases and Browse Sales screens. Search pages have different parameters to filter data. For instance throughout the “Browse Sales” page the user can search sales between dates according to sale date or delivery date.

The complete web site is shown schematically in Figure 4.10.

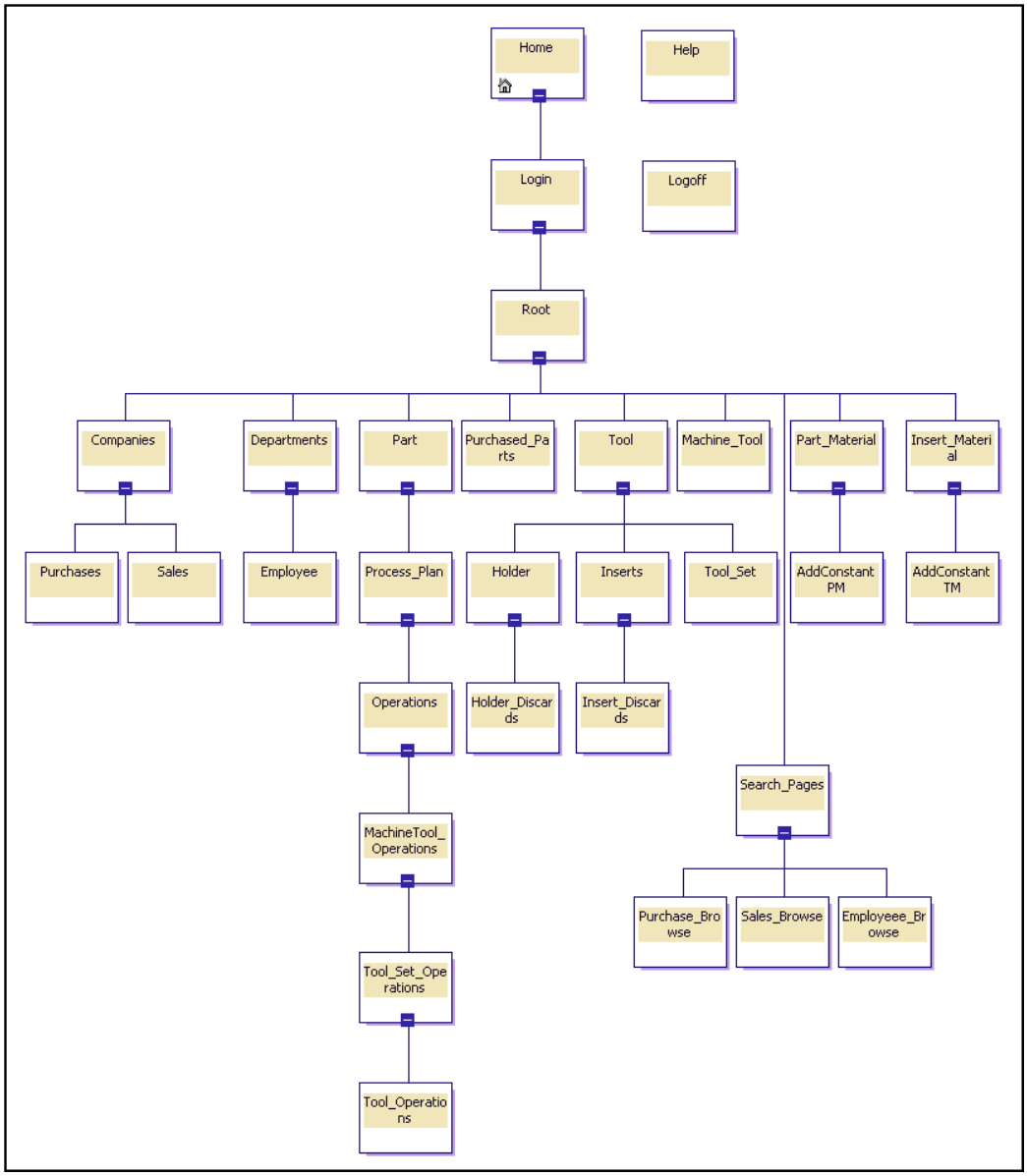


Figure 4.10 Site map

The web is published on Internet Information Server site at the address: http://cimlabserver.me.metu.edu.tr/SME. Information about "how to use the system" can be found at http://cimlabserver.me.metu.edu.tr/SME/Help.htm. Integrated Manufacturing technologies homepage can be reached from http://www.imtrg.me.metu.edu.tr.

CHAPTER 5

INTEGRATION OF THE AGENT AND OPTIMIZATION COMPONENTS INTO THE SME DBMS

5.1 Methodology

The “SME Database Management System” model and development phases had been presented in the previous section. This chapter focuses on the methodology of adding new applications to the system.

Several application programs may be used by several departments in an enterprise for different purposes. The persecution of the data flow is very crucial. If there are common data used by different programs and each program keeps the records of these data in different places some inconsistencies may occur. The solution is using a unique database for all of the user interfaces.

METUCIM is thought as an example enterprise and in order to provide data sharing and to eliminate update problems two separate applications “Optim” and “Agent”, using two separate databases are integrated into the “SME DBMS” thus a single database is constructed and all of the database objects used by the “Optim” and “Agent” applications in order to retrieve/add/delete/modify data, are redesigned to use this new database and integrated into the “SME DBMS”. Figure 5.1 shows the new system structure after the required modifications. The parts marked with “*” are the modified components of the other two applications. This new system structure facilitates the management of the data across the enterprise. The common data is defined once and is accessed and updated from all of the application programs. The system structure prevents waste of storage, eliminating data duplication and provides data consistency across the company.

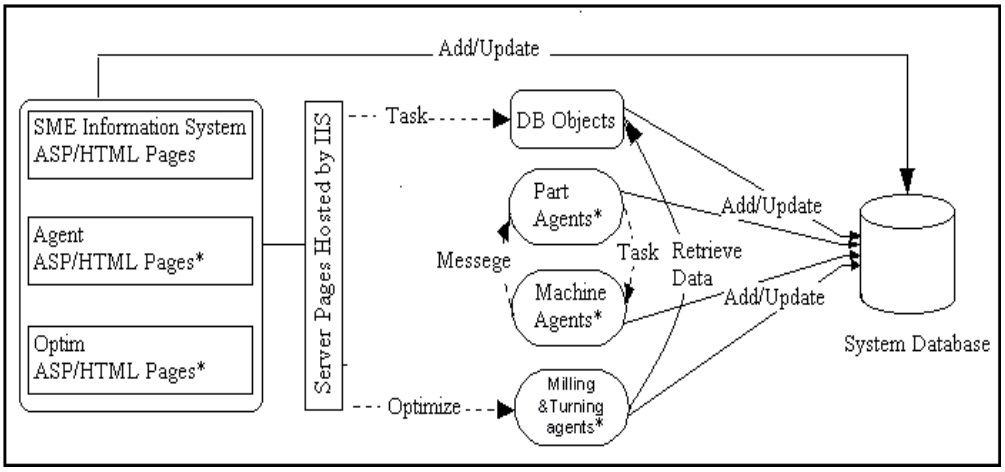


Figure 5.1 Entire System Components

5.2 Constructing the System Database

5.2.1 Integration of the Optimization Database into the SME DBMS Database

Optimization database, developed by Sarı [2], contains records used to calculate the optimum cutting conditions for single-pass milling and multi-pass turning operations with minimum production cost or maximum production rate using different optimization methods.

Different optimization techniques are defined in the database. The constants used for the calculations are stored in the optimization database together with the operational data. Tables of whose names identified with *"CON"* represent values of constants appearing in the optimization equations. On the other hand, tables of whose names started with "*LIST*" contain list of data related with part material, optimization, milling operation and machine tool.

The inputs such as work piece cutting, cost and time parameters and the calculated optimum cutting parameters are also kept in the related data tables so that browsing capability of these output values will be enabled for further analysis.

Figure 5.2 shows the original database diagram of the Optimization System Database developed by Sarı [2].

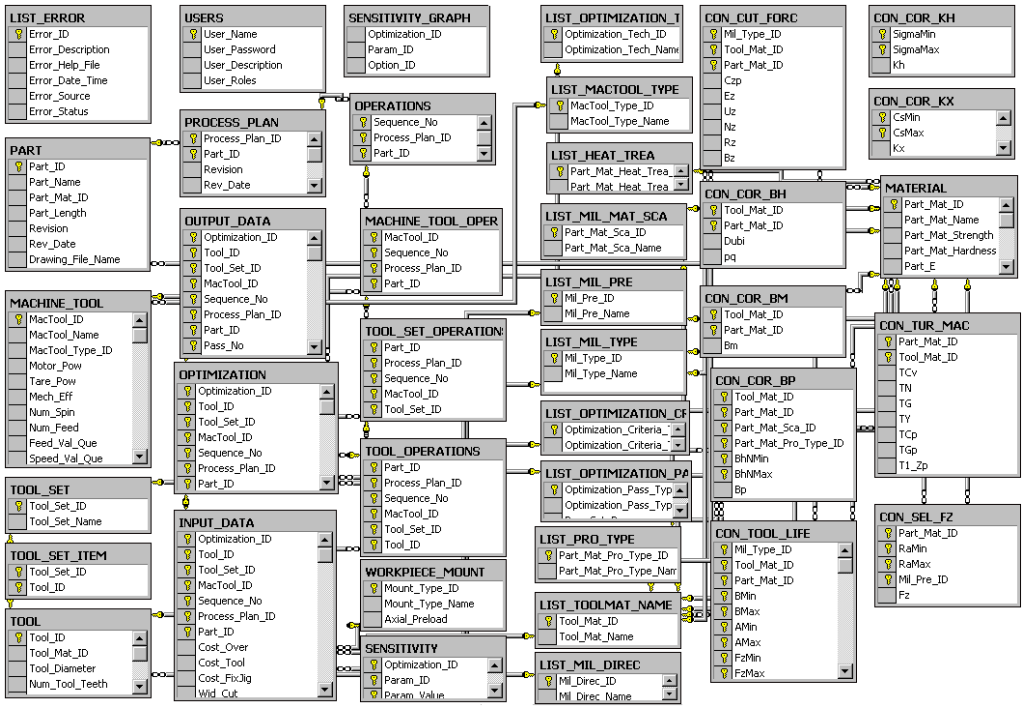


Figure 5.2 Database Diagram of the Optimization System Database [2]

In order to integrate the Optimization Database into the SME database, first of all the common data tables are designated. *Part, Machine Tool, Tool_Set, Tool_Set_Item, Users, List Error, Process_Plan, Operations, Machine_Tool_ Operations, Tool_Set_Operations, Tool_Operations, List_MacTool_Type, List_Mil_Mat_Pre, List_Heat_Trea, List_Pro_Type, List_ToolMat_Name* and *Material* tables also exists in SME database. The attributes of these tables are compared and differences are eliminated. For instance, the *Operations* table in SME database contains an additional attribute Person_ID defining the worker responsible for the operation while the Operations table in Optimization database does not. This additional attribute is a foreign key relating the table to the *Employee* table. As a consequence the *Operations* table of the new database has an additional attribute.

The most significant difference between two databases is that, the meaning of *Tool* tables in two databases is different. In fact the *SME.Inserts* table corresponds to the *Optimization.Tool* table. This difference leads to some structural changes. Because the Tool_ID is one of the main data of optimization system used in retrieving the parameters for several calculations. In the new database an Insert and a Holder constitute a Tool. The new structure is depicted in Figure 5.3.

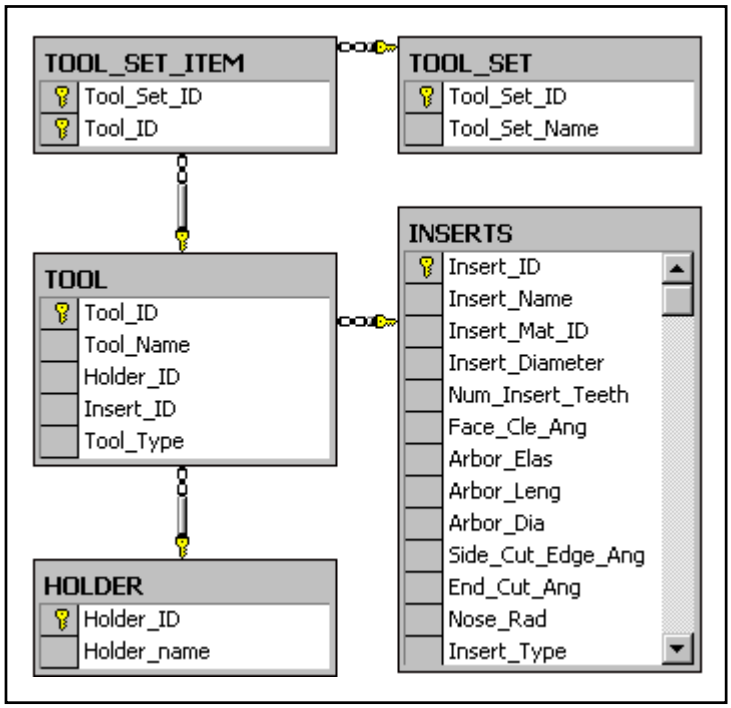


Figure 5.3 Modified Tool table and its relations

Also the stored procedures of the Optimization system are added to the SME database after some modifications.

The database structure after the Optimization database is integrated into the SME database is shown in Figure 5.4. The framed part in the figure shows the data tables of Optimization database that are added to the SME Database. A list of database tables and attributes are given in Appendix D.

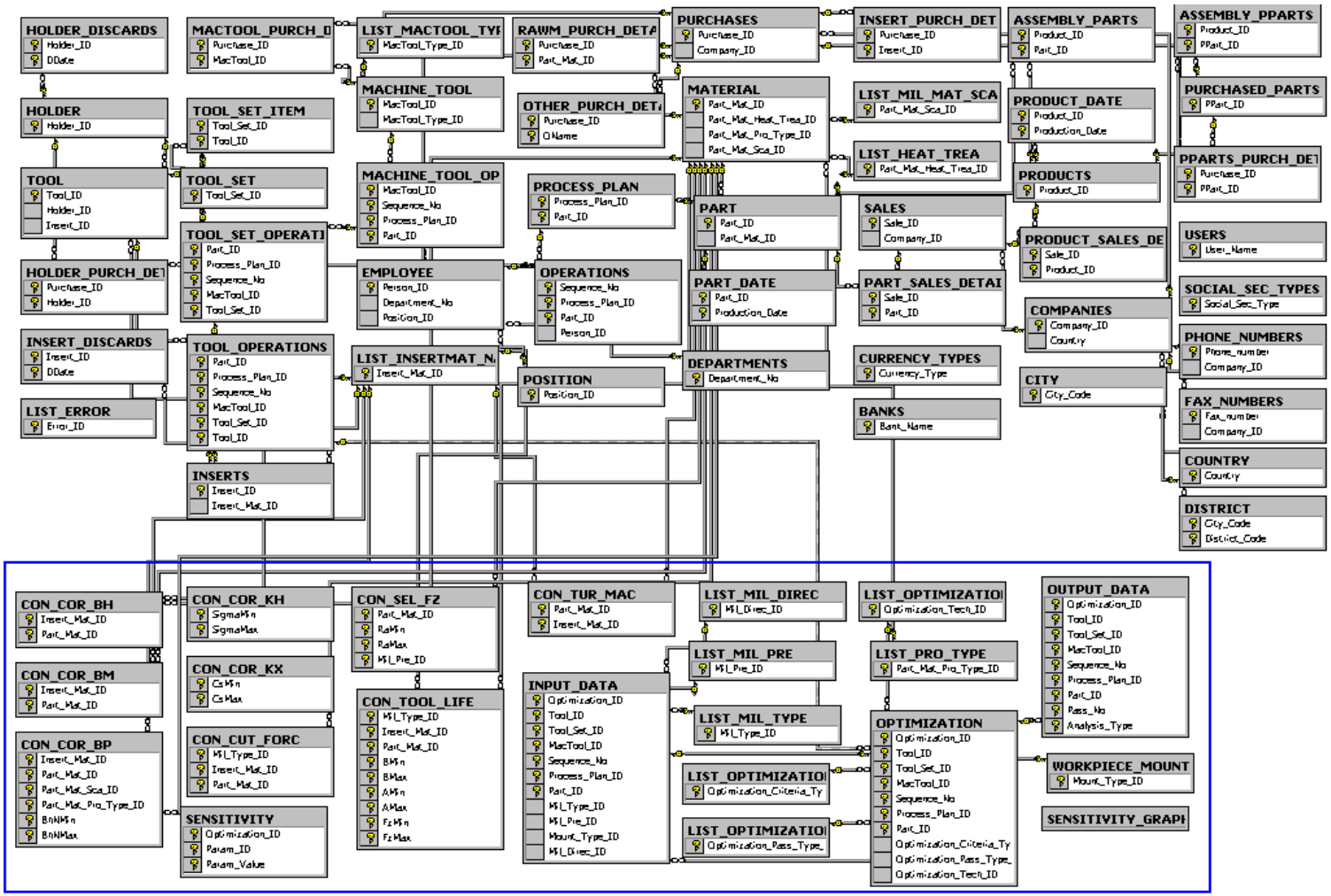


Figure 5.4 System Database after the Optimization Database is integrated into the SME Database

5.2.2 Integration of the Agent Database into the SME DBMS Database

Agent database, developed by Cangar [1], contains the records of a flexible manufacturing system using agent-based communications. Figure 5.5 shows the original database diagram of the Agent System Database developed by Cangar [1].

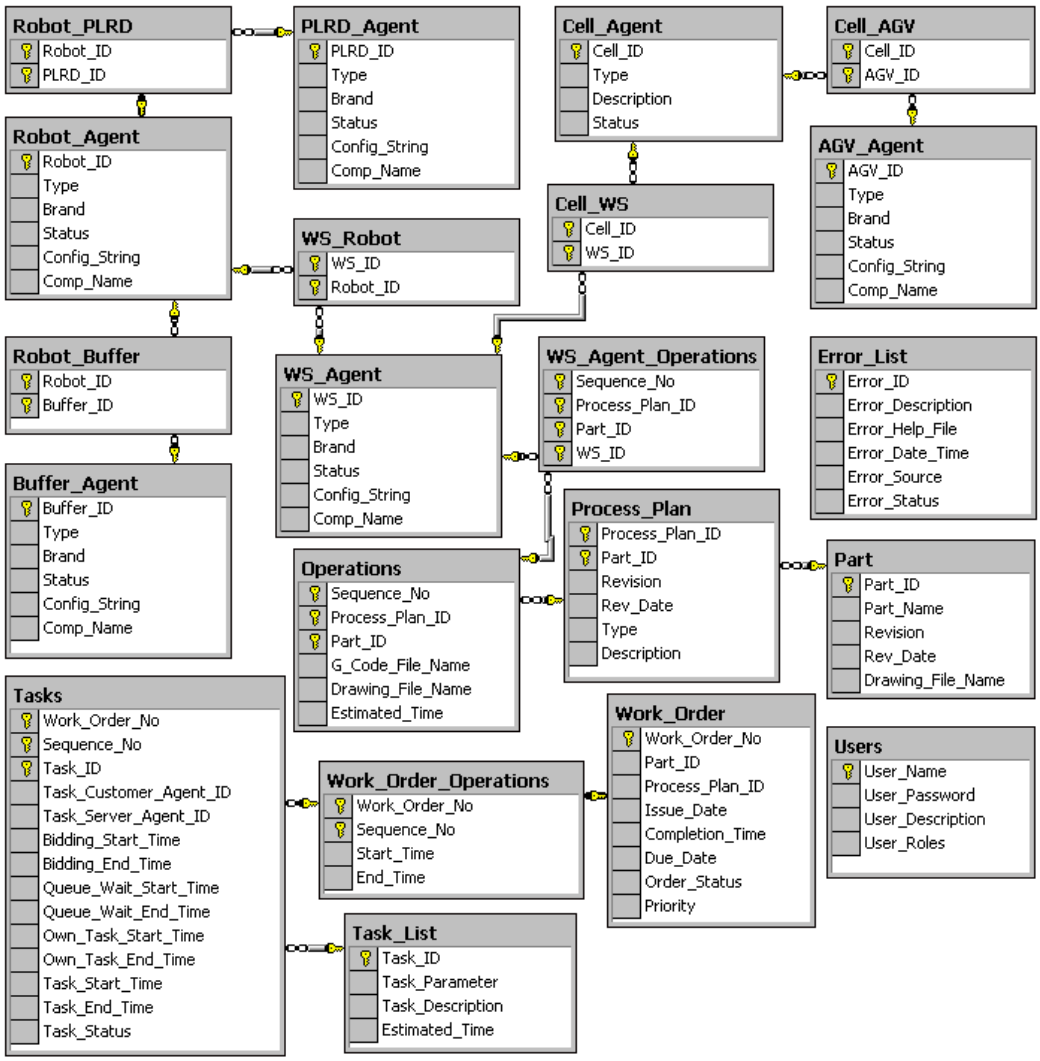


Figure 5.5 Database Diagram of the Agent System Database [1]

After the addition of the Optimization database to the system database, in order to integrate the Agent database, the common data tables are designated. *Part, Operations,* and *Process Plan* are the common data tables of the two databases. The attributes of the common tables of two databases are compared and attributes that are different are designated. Figure 5.6 shows the *Operations* table and its attributes in two databases as an example.Operations table in SME database have additional attributes besides the common attributes. Thus the new system's *Operations* table is selected as *SME.Operations* table. And the WS_Agent_Operations table is related to the SME.Operations table. The same methodology is followed for the rest of the data tables.

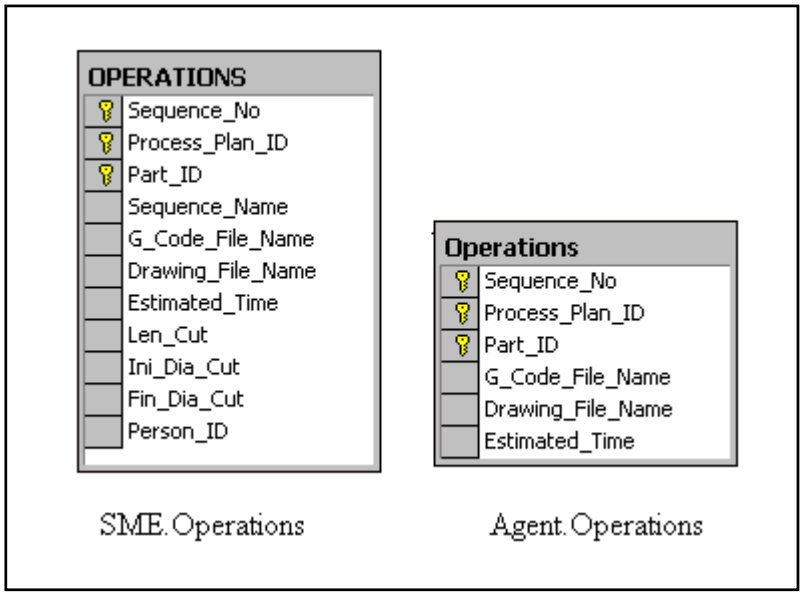


Figure 5.6 SME.Operations and Agent.Operations Tables

Figure 5.7 shows the system database after the Agent database is integrated. The framed part in the figure expresses the data tables of the Agent system that are added to the SME Database. A list of database tables and attributes are given in Appendix D.

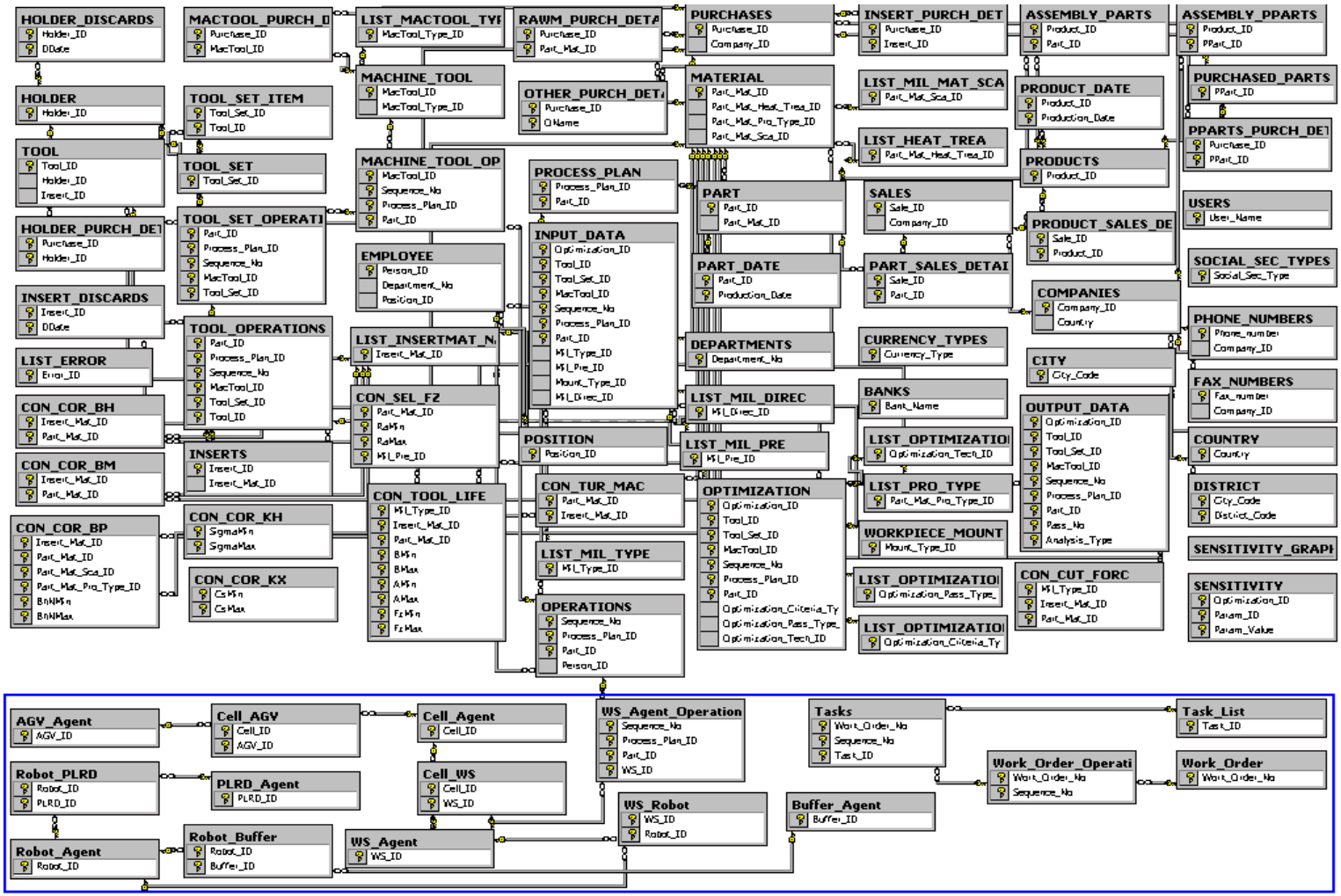


Figure 5.7 System Database after the integration of Optimization and Agent Databases into the SME Database

5.3 Modifications done in Optimization Components

The application program “Optim” was using the Optimization database before the construction of the new system database. As several modifications are done in the database structure while integrating it into the SME DBMS database, some changes are inevitable in the business and presentation services components used by the “Optim” application program in order to provide conformity with the new database.

The database objects, created before, are modified to use SME database instead of Optimization database and are included to the business tier of the “SME DBMS”. As it is mentioned before almost each table in the database has a corresponding database object. The database objects, written for Optimization that do not exist for SME Information System are added to the Visual Basic project and compiled as DLLs.

The Turning and Milling agents in Optimization system were modules used to calculate optimization results using different data from different tables. Optimization. Tool table was containing the insert data. But the new developed database contains this information in Inserts table so in order to take insert data, the Insert_ID is found using the Tool_ID value. The example code from SME.Milling_Agent is given in Appendix C Section C.9.

After the required modifications are done in database objects and agents, the class modules are recompiled as COM DLLs. An overview of the MTS screen containing objects that are used by “SME Information System” and “Optim” applications is shown in Figure E.1 in Appendix E.

After recompiling the database objects, the required modifications in the presentation services of the Optimization system are determined. Two new ASP pages for *Holders* and *Tools* are added, the existing *Tool* page is renamed as *Inserts* page and the related codes are modified. The “Optim” uses five small programs for the graphical demonstration of the optimization results: Sensitivity Package, Turning Package, Milling Package, Tradeoff Package and Agent Package. The codes of these programs are modified according to the new database

structure and recompiled. The codes of the ASP pages used for graphical representation are also modified.

5.4 Modifications done in Agent Components

The database objects and Part Agents used by the "Agent" application program are modified to use the new constructed database and after recompiling they are added to the business tier of the "SME DBMS". The MTS screen containing the whole system components is shown in Figure E.2 in Appendix E.

Eight agent controller EXE programs to drive the hardware components are modified according to the new database and system components. These programs are: AGVOcx.Exe, CMMOcx.Exe, CncOcx1.Exe, CncOcx2.Exe, CncOcx3.Exe, ConveyerOcx.Exe, PLRDOcx.Exe, RobotOcx.Exe.

The ASP Pages of "Agent" application program are modified and the MTS objects of SME package are used instead of Agent package throughout the pages for data retrieval, addition or modification.

CHAPTER 6

TEST RUNS

6.1 First Test Run

The first test run shows steps of defining operational data of a part from the web based user interface “SME Information System”.

The main entrance of the system is from Login.asp. An ordinary user will login with Name: user, Password: guest, he/she can gain the access based on "User". If you have the username and password based on "Engineer" or “Admin”, you can delete, add, modify any information contained in this site. Figure 6.1 shows the login screen. The user rights are based on “Admin” in this test run.

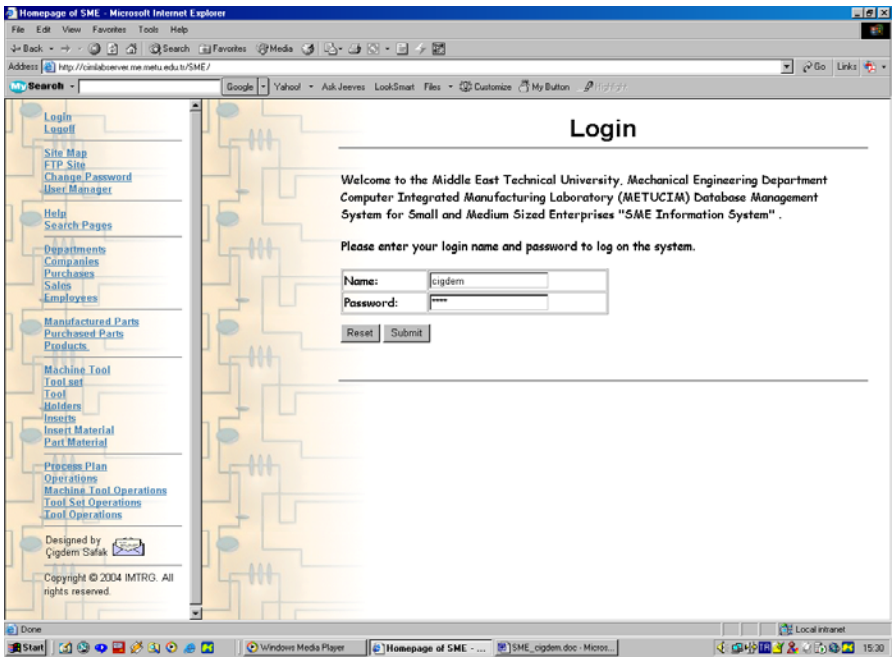


Figure 6.1 Login Screen

After logging in, the Root window containing a list of links is opened. These links are classified as it is shown in Figure 6.2 in order to provide a user friendly environment.

All of the links can be also found on the left frame of each asp page. It is assumed that the available static Machine Tools, Tool Sets, Tools, Inserts and Holders are defined before the test run from the related pages. The general rule for defining a new equipment is to click the *Add* button and after entering the required data to click on the *Submit* button in order to record the data to the database.

In this test run first of all the Part that will be manufactured is defined from the "Manufactured Parts" window.

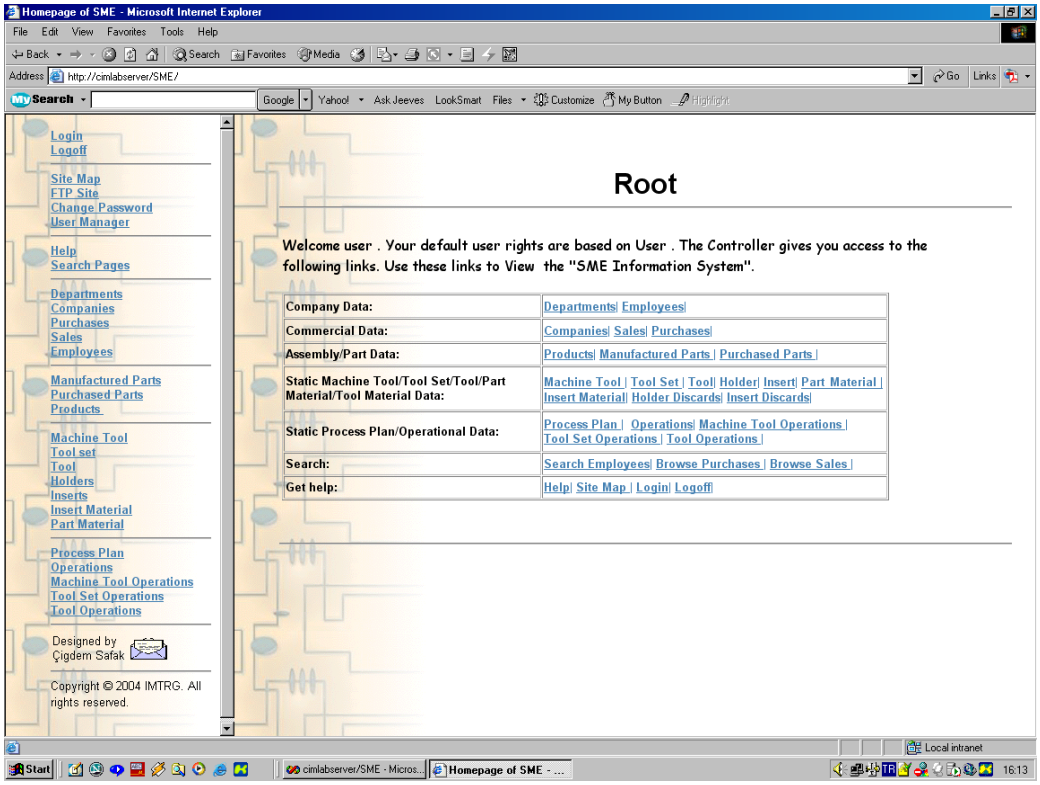


Figure 6.2 Root Window

Part_ID, Part Name, Part Length, Revision, Revision Date, Raw Part Length are defined throughout the "Manufactured Parts" window as it is shown in

Figure 6.3. The technical drawing of the part is addressed by adding the URL address of the drawing file. Also the production dates and number of produced parts can be defined for each part. Part Material is selected from the related combo box among the available part materials that are previously inputted from the "Part Material" page. Part material data can be viewed clicking on the *Browse Part Material* buttton. Figure 6.3 shows the "Manufactured Parts" screen after submitting Part 10001 to the database.

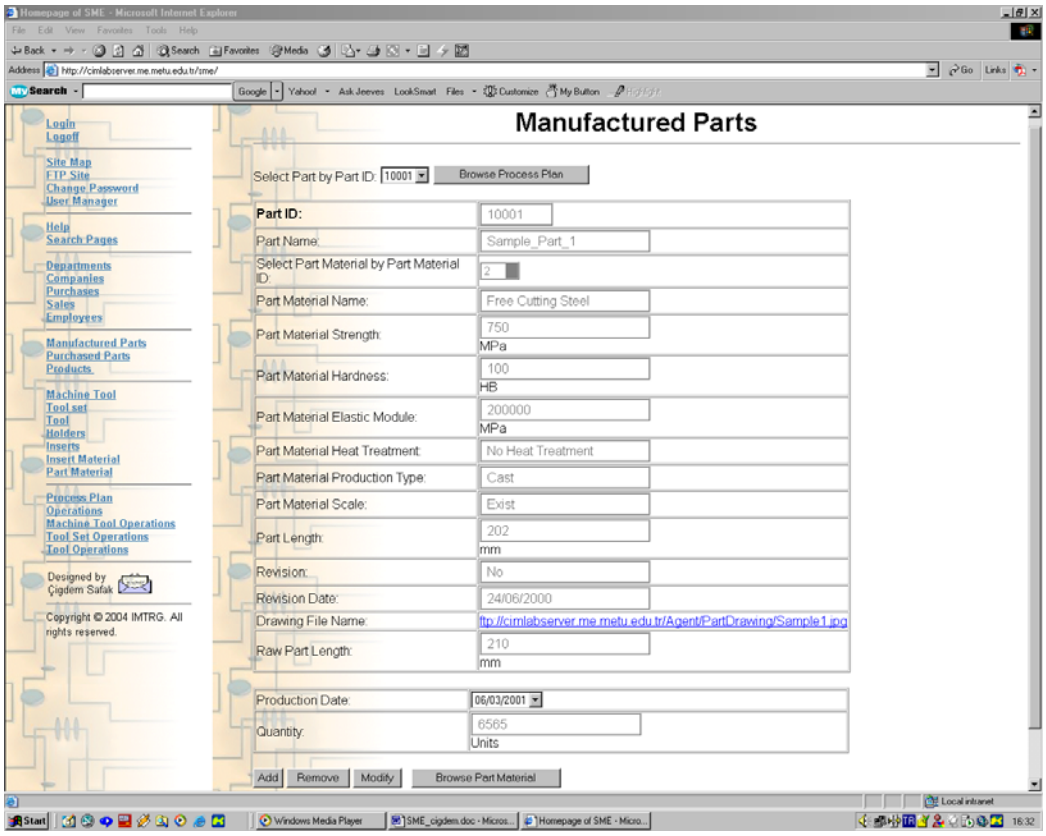


Figure 6.3 Manufactured Parts Window

The first step after identifying the part properties is to define the process plan for the part. After clicking on the Browse *Process Plan* button the "Process Plan" window opens containing the available process plans of the selected part.

The “Process Plan” window is depicted in Figure 6.4. Each part can have several process plans and in this example Part 10001 has two process plans 20001 and 20002. The Revision, revision date, revision type and description are the data identifying a process plan. *Browse Part* button enables user to view the selected part’s properties from “Manufactured Parts” window.

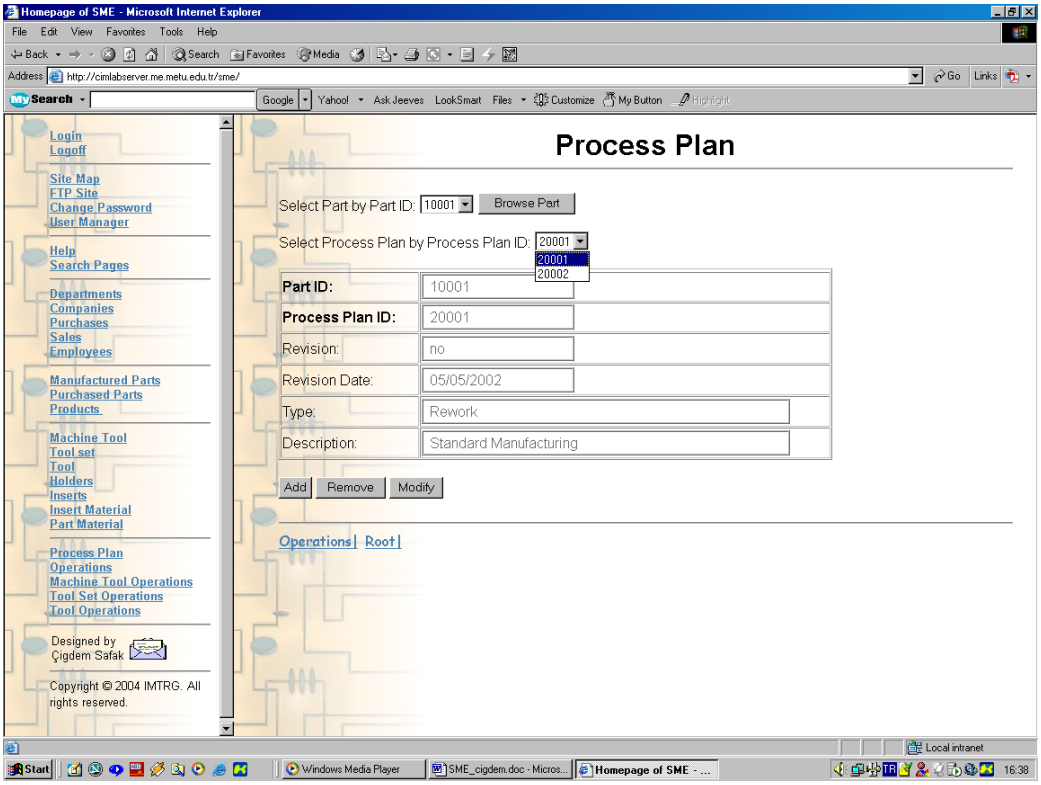


Figure 6.4 Process Plan Window Browse Screen

A new process plan with an ID 20003 is added for Part 10001 in order to show the planning of the operational steps. The Figure 6.5 shows the Add screen of the Process Plan window. The new Process ID is given automatically by the system but it should be mentioned that this ID is not saved to the database immediately. If at the same time an other user tries to input a process plan data to the system, and approves it before, the new defined process plan will be recorded with an ID of 20004. After entering process plan data to the related

fields, the data is added to the database by clicking on the *Submit* button and the screen switches to the Browse mode.

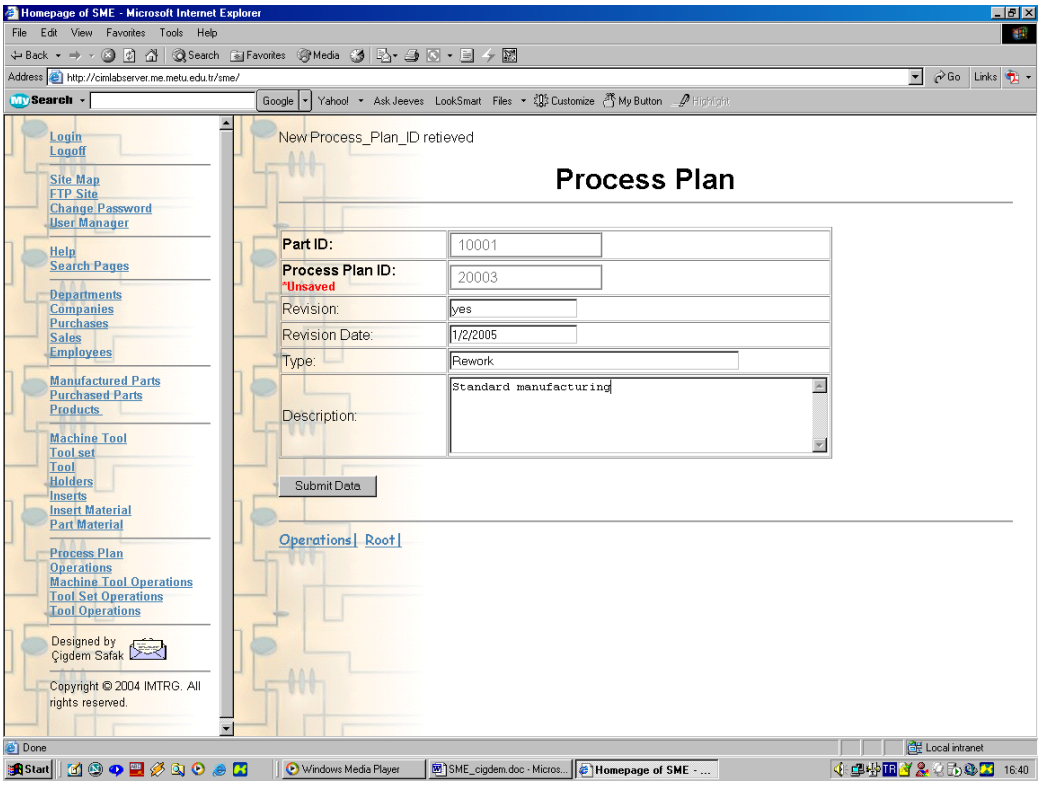


Figure 6.5 Process Plan Window Add Screen

After defining the new process plan for the selected part, the main operational data is defined from the "Operations" window. Several sequences for an operation can be added from this window. Each sequence is defined by the operation type (turning or milling), the operator, estimated process time and the length of cut. Initial and final diameters of the work piece is defined for turning operations. In this example a turning operation for Part 10001 with a Process Plan 20003 is added. Figure 6.6 shows the inputs of the operation. All of the operation sequences are added throughout this window.

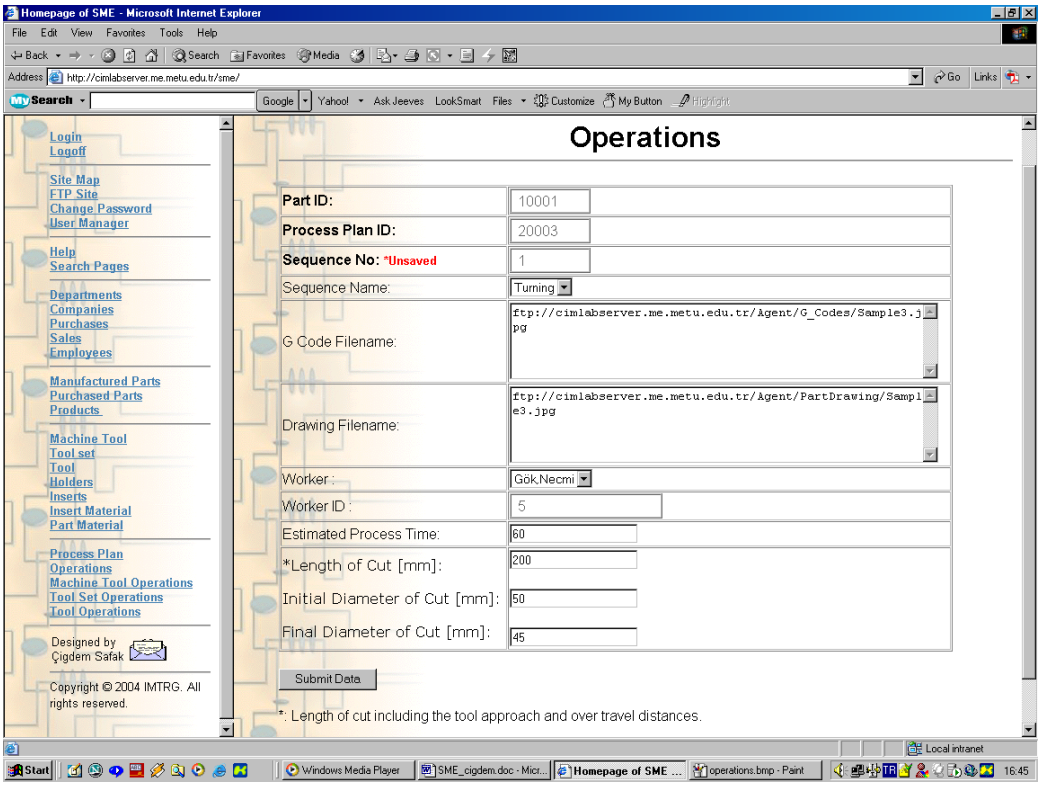


Figure 6.6 Operations Window Add Screen

The next step is choosing the machine tool for the defined operation sequence. As it is shown in Figure 6.7 only the machine tools that are suitable for the defined operation type are populated to the combo box. Machine Tools with IDs 30002 and 30004 are the turning machine tools available in the system database. In order to view the properties of the selected machine tools the user should click on the *Browse Machine Tool Properties* button. In this test run Machine Tool 30002 is selected and assigned for the first sequence of the operation. The state "Assigned" means that the selected machine tool is or has become a server of the indicated operation. For each sequence of the operation a machine tool should be assigned in order to define the operation completely. *Browse Operation* and *Browse Process Plan* buttons enable user to access the Operations and Process Plan windows for the related data respectively.

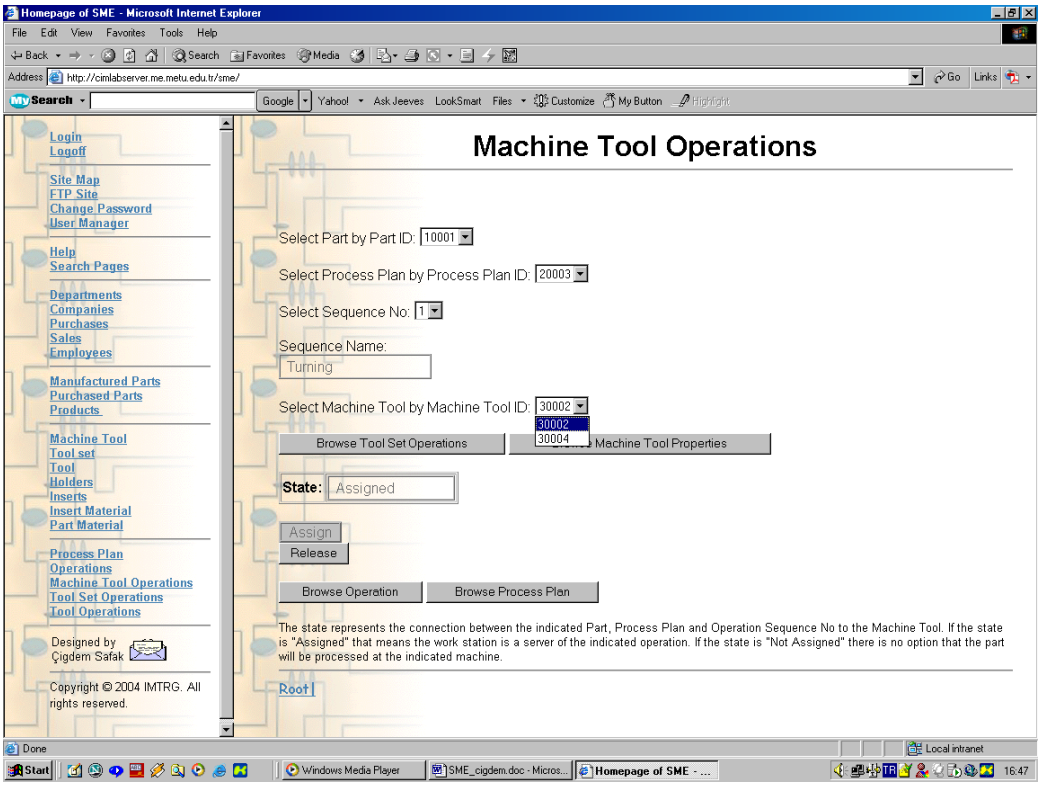


Figure 6.7 Machine Tool Operations Window

After assigning the machine tool for the first sequence of the operation, the tool set is chosen. The appropriate tool sets that are previously defined as turning tool sets from the "Tool Sets" window are defined with IDs 40001 and 40003 as it is shown in Figure 6.8. In order to view or change tool set properties *Browse Tool Set Properties* button can be used. In this test run Tool Set 40001 is selected and assigned for the first sequence of the operation. The state shows if the tool set is assigned or not. In order to assign the selected tool set for the operation the user should click on the *Assign* button. *Browse Operation, Browse Process Plan* and *Browse Machine Tool Operations* buttons are added to the window in order to ease the access to the previous screens and enable user to change required data at every step. In Figure 6.9 the properties of Tool Set 40001 can be viewed.

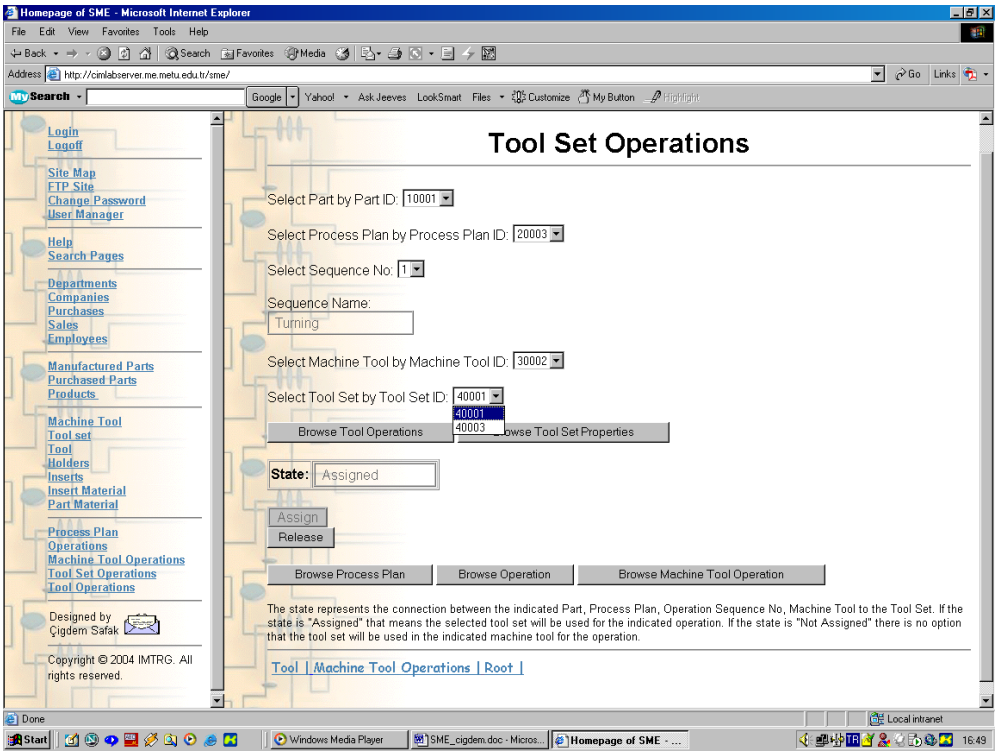


Figure 6.8 Tool Set Operations Window

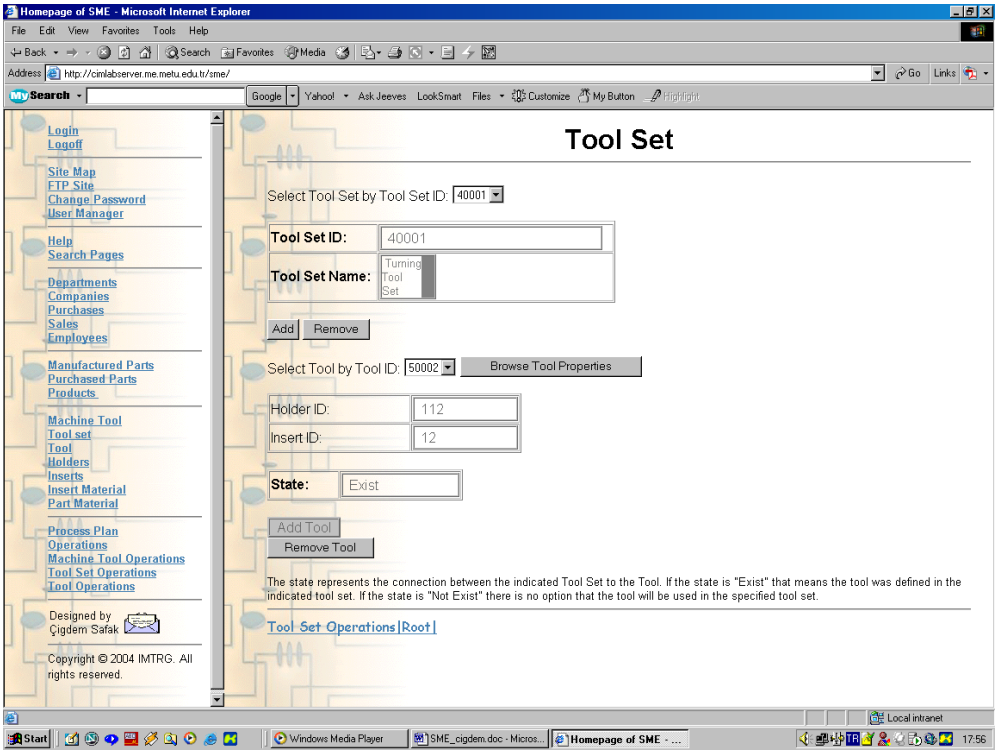


Figure 6.9 Tool Set Window showing the properties of Tool Set 40001

The last step of defining an operation is assigning the tools that are going to be used for the operation. The “Tool Operations” window is depicted in Figure 6.10. The tools combo box contains only the tools defined for the selected Tool Set 40001. The user can assign one or more tools for the operation. Before selecting the appropriate tools the properties of each tool can be viewed throughout the ”Tools” window by clicking on the *Browse Tool Properties* button. In this example Tool 5002 is selected for the operation.

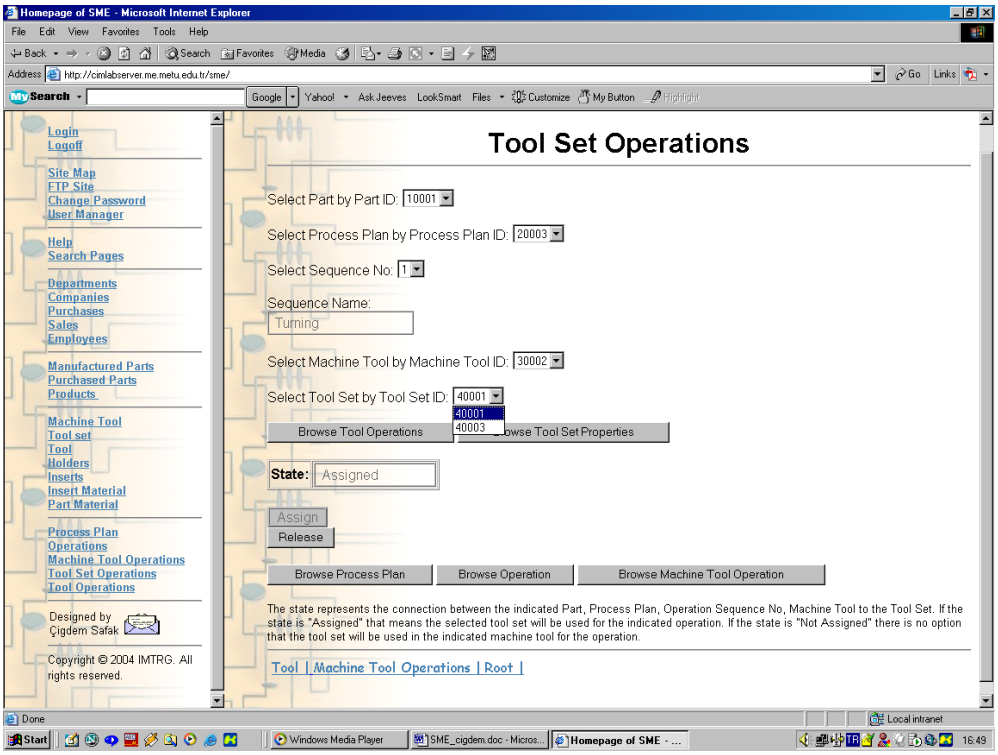


Figure 6.10 Tool Operations Window

6.2 Second Test Run

The second test run shows the steps of defining the sales data of the SME. The enterprise sells parts and products. Adding a part to the information system has been shown in the previous section. Figure 6.11 shows browse screen of the

"Purchased Parts" window after the addition of the Purchased part defined by ID 70001.

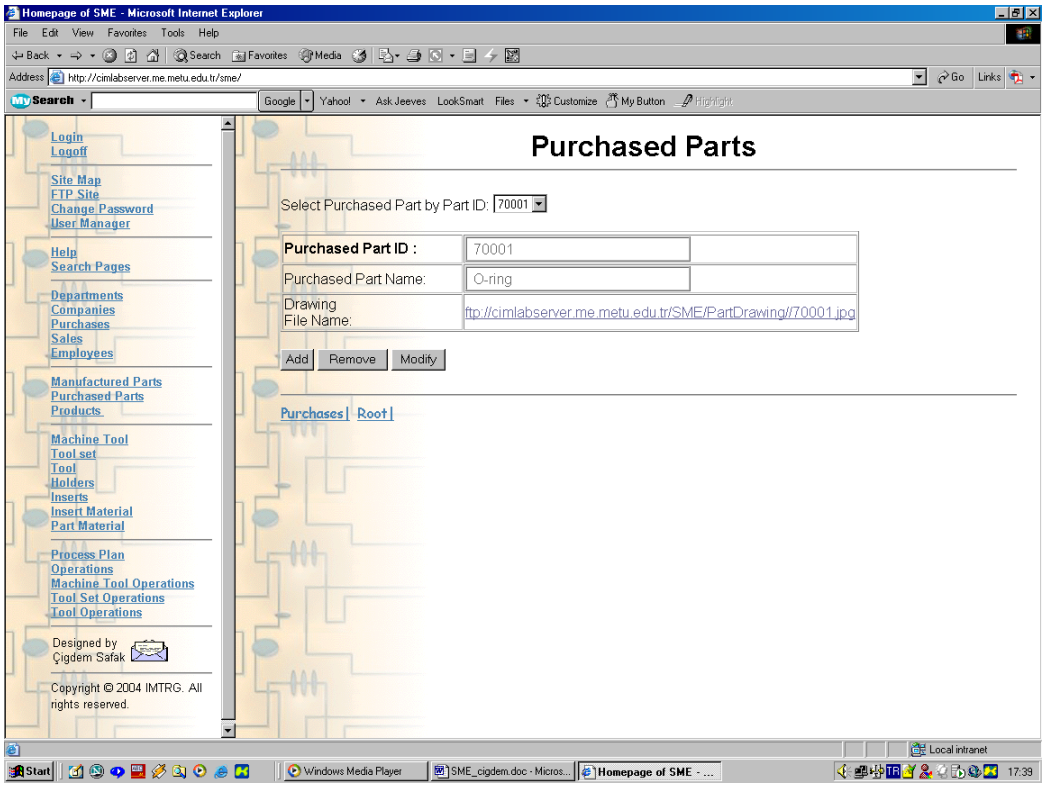


Figure 6.11 Purchased Parts Window

The products are assemblies that are consisting of several manufactured and purchased parts. The "Products" window consists of two sections. The Product_ID, Product Name and the URL address of the drawing file of the product is defined in the first section. The user should approve this general information before inserting data to the second part. After the data is recorded the second part of the window becomes enabled. Figure 6.12 shows the browse screen of the "Products" window after the addition of Product 3001. The user can define one or more assembly dates for a product, together with its quantity. Manufactured and the purchased parts of an assembly are selected from the related combo boxes

among the parts that are available in the system database. How many of them are used for the assembly are also defined by the user.

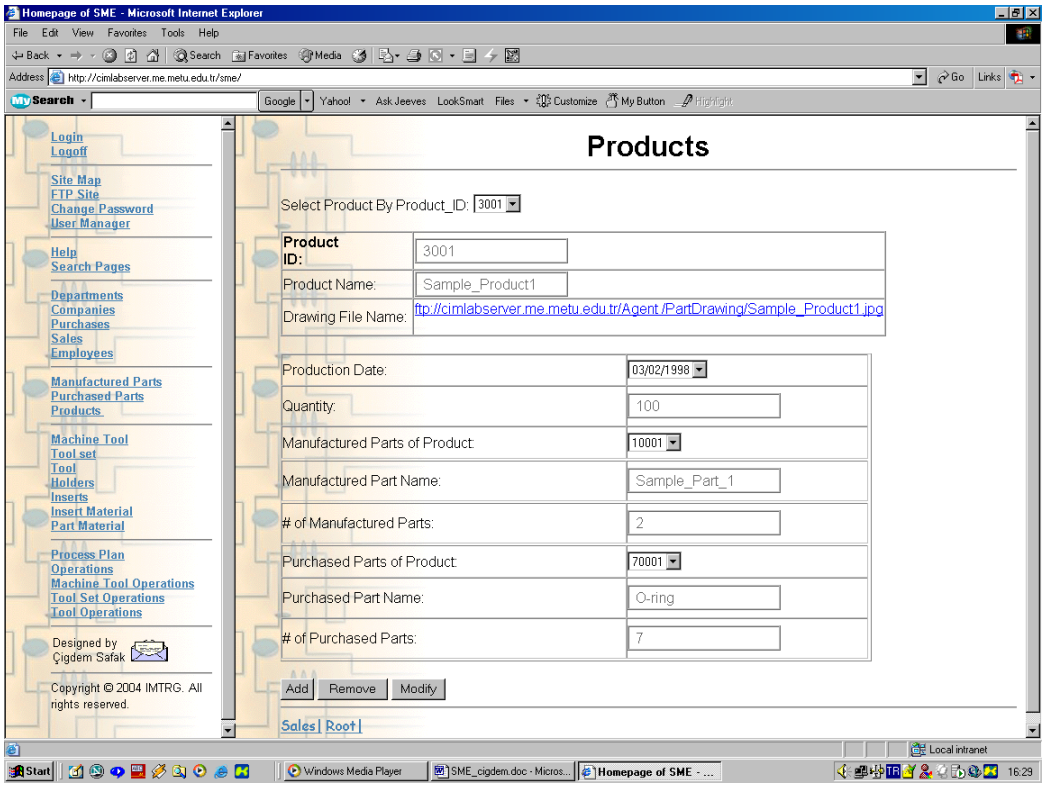


Figure 6.12 Products Window

In order to define sales, the companies commercially related to the SME should be added to the system database. Figure 6.13 shows the screen after the addition of the company called DECON A.Ş. Company ID, Company Name, Country, Address, City, District, Postal Code, Tax Office, Tax No, Bank, Branch, Bank Account No, E-mail, Web site, Contact person, phone and fax numbers of DECON A.Ş. are defined.

In order to generate company report in MS. Word the user should click on the *Generate Company Information Form* button. Figure 6.14 shows the generated report.

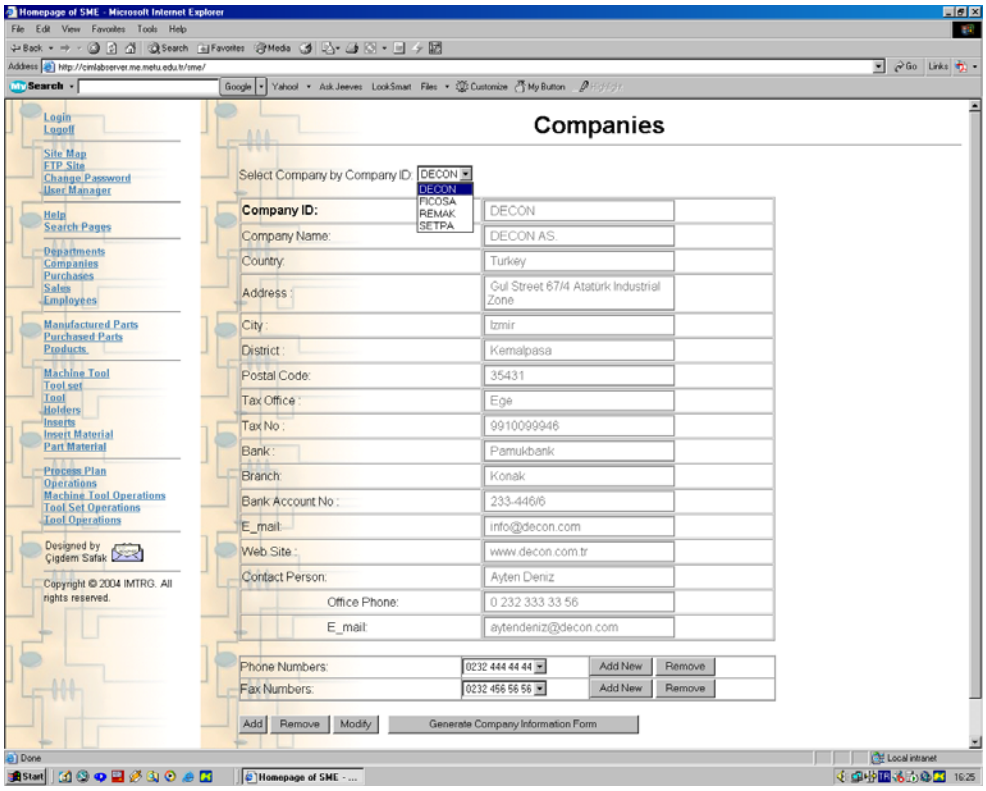


Figure 6.13 Companies Window Browse Screen

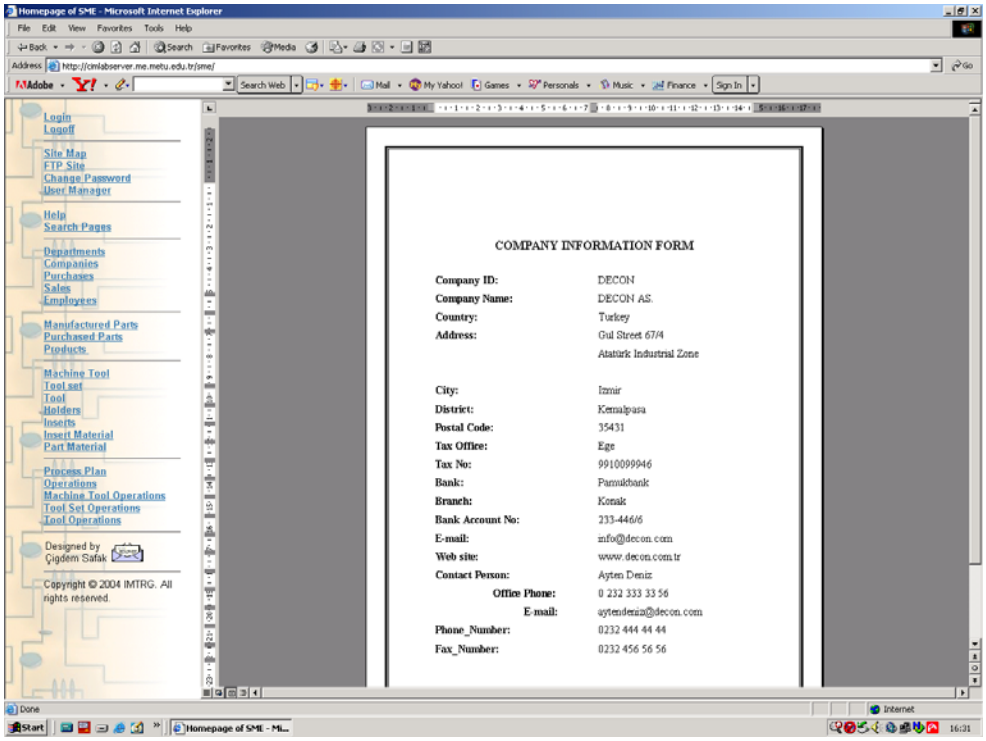


Figure 6.14 Generated Company Information Form

Using the Sales link at the bottom of the window the “Sales” window is opened in browse mode showing the existing sales in the system database. In order to add a new sale the user should click on the *Add* button and switch to the add mode as it is shown in Figure 6.15. The following screen consists of two parts that is addition of a new sale to the system is achieved in two steps. The second part becomes enabled after submitting the data entry of the first part by clicking *Submit New Sale Data* button. The first part contains general information about the sale. In this test run sale with an ID 6 is added to the system. The related company is selected from the combo box among the companies that are previously recorded to the database. After inserting Invoice no, Date, Delivery address, Delivery date, Payment type and Payment Term the first step is completed by submitting the data.

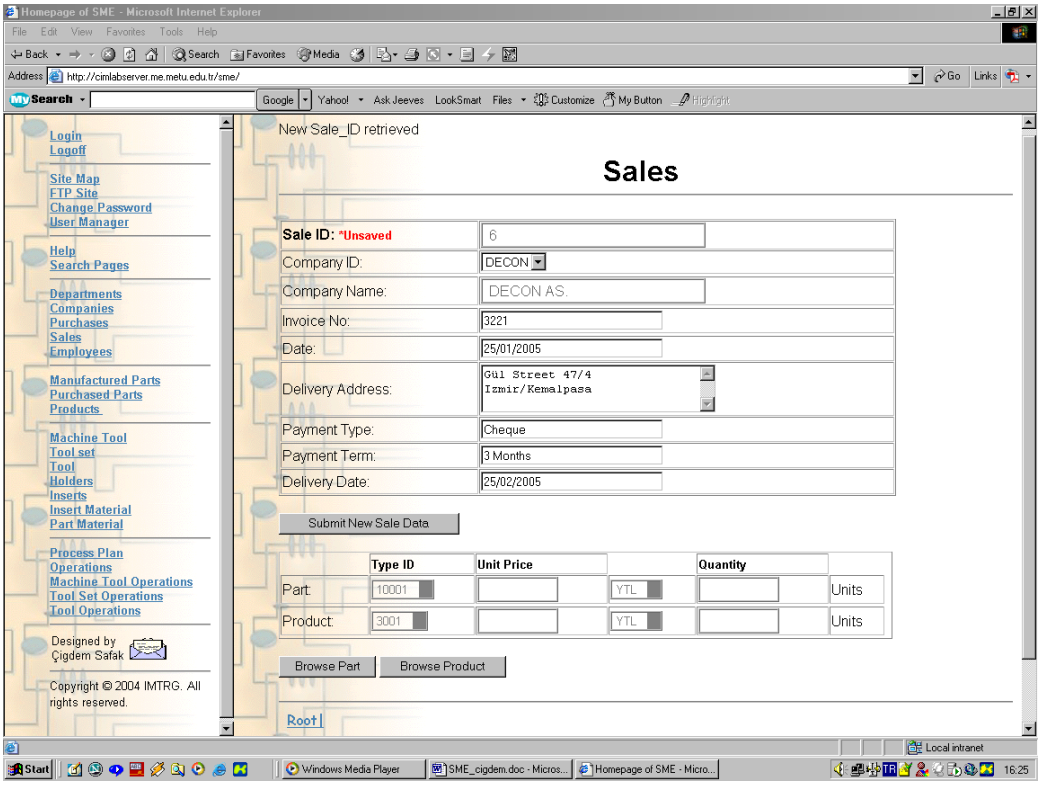


Figure 6.15 Sales Window Add Screen

Before submitting general sale data in the first part

Second part contains details about the sold parts and products. After the part and/or product are selected from the combo boxes and the unit price and quantity of the selected item is entered to the related fields, the data is recorded to the system database by clicking on the save sign. Several products or manufactured parts can be added to a sale by repeating the same action. In this test run Part 10001 and Product 3001 are added to the sale. *Browse Part* and *Browse Product* buttons give link to the related pages containing the data of the selected item. The addition of the new sale is completed successfully.

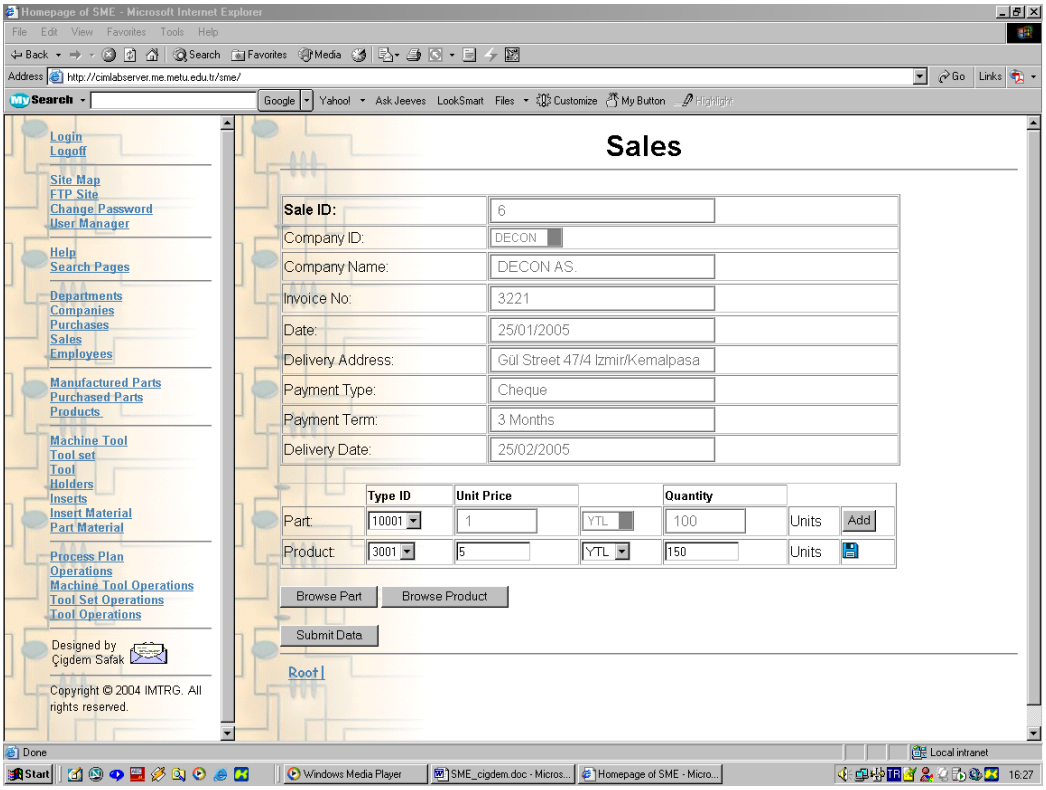


Figure 6.16 Sales Window Add Screen
After submitting general sale data in the first part

There are search pages in the "SME Information System" enabling the user to filter the search according to a selected parameter. Figure 6.17 shows the

search window for sales. Search window for sales contains six different types of filtering parameters. In the figure the parameter is Company ID. Search results for the sales related to the DECON AŞ. are shown as a table at the bottom of the screen.

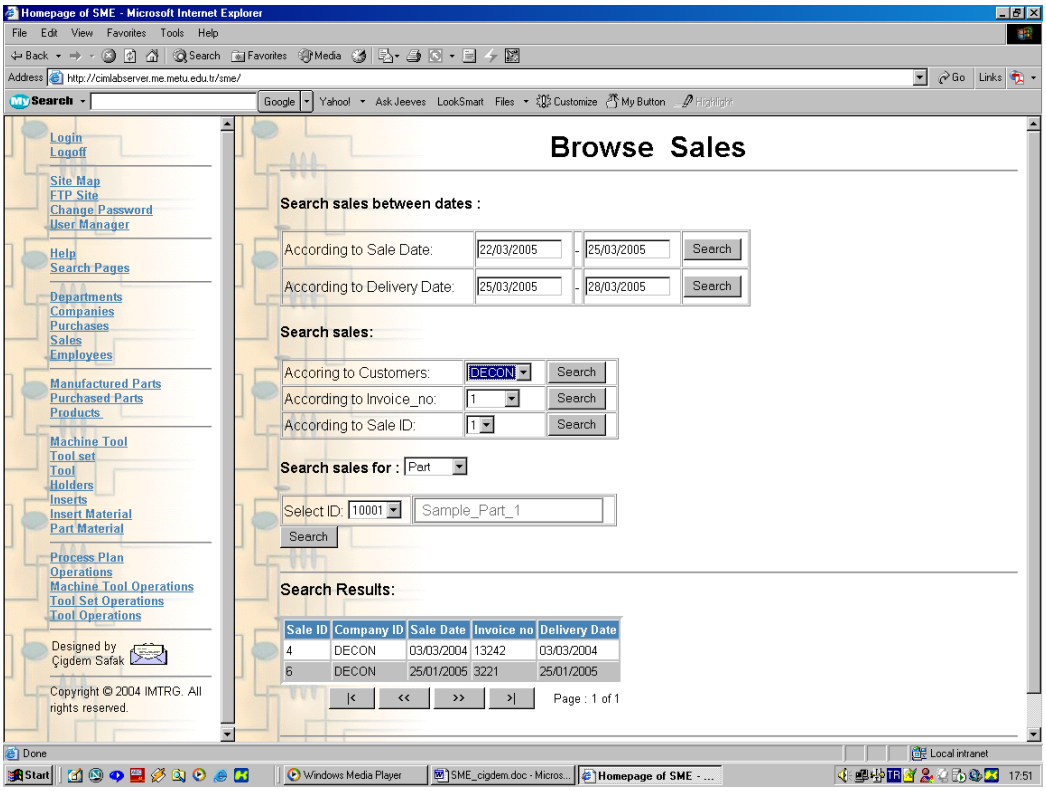


Figure 6.17 Browse Sales Window showing the search results according to the DECON A.Ş. Company

CHAPTER7

CONCLUSION AND FUTURE WORK

In this study a unified web based information system for small and medium sized enterprises is developed, having a unique database and three application programs. The effective features and tools of Windows DNA and the integration definition for information modeling (IDEF1X) standards are used to design the system components. The developed system enables user to define and store all of the enterprise data, optimize cutting conditions for single-pass milling and multi-pass turning operations, and control the shop floor operations from web throughout three application programs using SME DBMS Database. Using a unique database for all applications eliminates data redundancy and provides data sharing across the enterprise.

Database Management System (DBMS) is the most significant tool developed to serve multiple users in a database environment consisting of programs that enable users to create and maintain a database. Today, SMEs and their requirements are rapidly changing so it is crucial to recognize that the management of an enterprise efficiently depends on the efficient use of knowledge. Creation of information systems specific to the enterprise demands enables enterprises to manage and disseminate data correctly without redundancy.

In this thesis an attempt is made to integrate “Optim” and “Agent” applications into a unified system and for this, data flow diagrams are constructed in order to constitute a basis for the system design after determining the requirements of a manufacturing company. A new web based application namely “SME Information System” is developed. Bearing in mind that the system should be designed in conformity with the “Agent”, an agent based flexible manufacturing cell controller, and the “Optim”, a web based software for

optimizing cutting parameters, the new database management system is implemented for these three application programs running off a unique database.

“SME Information System “ provides newly developed user interface enabling storing and retrieving up-to-date, accurate and relevant financial, operational and institutional information for the user. Organizing the resources facilitates the persecution of data flow and information retrieval. The main objective is to establish an information chain, enabling user to follow processes performed within the enterprise from production to trading. Throughout the ASP pages the operation steps of a part together with the process plans, utilized machinery and equipment can be defined by the authorized users.

The Agent and Optimization systems components are modified and integrated into the new DBMS in order to prevent data duplication and facilitate retrieval of up-to date information within the enterprise.

The scope of the study extends from job shop planning to shop floor control with online browsing capabilities from the widespread Internet environment, equipped with a user-friendly interface. However, the following possible future improvements can be specified:

1. The system database can be extended enabling the addition of new modules to the “SME Information System” application program for cost estimation, accounting and inventory management.
2. Reporting can be enhanced. Regular system report templates can be prepared according to the demands of the departments. Storing and printing of reports can be automated throughout the application programs.
3. Error handling and recovery functions should be improved and coordinated to the use on ASP pages.
4. The Internet pages can be enhanced with Computer Aided Design (CAD) and Computer Aided Process Planning (CAPP), and G-Code Generation to form an integrated peripheral of CAD and job shop planning.

REFERENCES

[1] Cangar T., 2000, “Development of an Agent Based Flexible Manufacturing Cell Controller Using Distributed Internet Applications”, Master Thesis Middle East Technical University Ankara Turkey, pp. 91

[2] Sarı B., 2000, “Development of a Job Shop Planning System: Optimization of Machining Operations using Windows DNA Architecture”, Master Thesis Middle East Technical University Ankara Turkey, pp. 104

[3] Latisen G., Vossen G., 1998, “Models and Languages of Object Oriented Databases”, Addison Wesley, pp.4,7

[4] Kroenke D.M., 2000, “Database Processing Fundamentals, Design and Implementation “, Seventh Edition, Prentice Hall, pp. 273-288

[5] Elmasri R. A., Navathe S., “Fundementals of Database Systems”, Third Edition, Addison Wesley , pp.4,8-11,57

[6] Pratt P. J., Adamski J. J., 1991, “Database Systems Management and Design”, Second Edition, Boyd and Fraser Publishing Company, pp. 13-14

[7] Silberschatz A., Korth H. F., Sudarshan S., 1997, “Database System Concepts”, Third Edition, McGraw-Hill, pp.253-264, 547

[8] Özsu T. M., Valdurez P., 1999, “Principles of Distributed Database Systems”, Second Edition, Prentice Hall, pp.562.

[9] Harris D., 2003, “Systems Analysis and Design for the Small Enterprise”, Third Edition, Thomson Learning, pp. 417

[10] MacFadden F. R., Hoffer J. A., Prescott M. B., 1999, “ Modern Database Management”, Fifth Edition, Addison Wesley, pp. 298, 302-304

[11] Harris S.E., Katz J. L., 1991, “Organizational Performance and Information Technology Intensity in the Insurance Industry”, Organization Science (2:3), 1991, pp. 263-295.

[12] Mitra S., Chaya A. K., 1996, “Analyzing Cost-Effectiveness of Organizations: The Impact of Information Technology Spending,” Journal of Management Information Systems (13:2), 1996, pp. 29-57.

[13] Johnson R., Lawrence P. R., “Beyond Vertical Integration: The Rise of the Value-Adding Partnership” Harvard Business Review, July-August 1988, pp. 94-101..

[14] Simon, “A Behavioral Model of Rational Choice,” Quarterly Journal of Economics (69), February 1955, pp. 99-118.

[15] Zmud R.W., and L.E. Apple, “Measuring Technology Incorporation /Infusion,” The Journal Of Product Innovation Management (9:2), 1992, pp. 148-156.

[16] Cash J. I., and Konsynski B., “IS Redraws Competitive Boundaries,” Harvard Business Review (64:2), 1985, pp. 134-142.

[17] Boudreau M. C., Robey D., (1999), “Organizational Transition to Enterprise Resource Planning Systems: Theoretical Choices for Process Research”,

Proceeding of the 20th international conference on Information Systems, 291-299.

[18] Klaus H., Rosemann M., Gable G. G., 2000, "What is ERP?", Information Systems Frontiers 2:2, 141-162.

[19] Russell R., and Taylor B. W., 1995, "Production and Operations Management: Focusing on Quality and Competitiveness", Englewood Cliffs, NJ: Prentice-Hall, Inc

[20] Sheikh K., 2003, "Manufacturing Resource Management (MRPII) with an introduction to ERP, SCM and CRM", McGraw Hill, pp. 495.

[21] Glass R. L:, 1998, "Enterprise Resource Planning: Breakthrough and/or Term Problem?", Data Base (29:2), pp.14-15

[22] Markus M. L, 1997, "The Qualitative Differences in Information Systems Research and Practice", Chapman and Hall, pp.11-27.

[23] Dicarerino A., Larsen K., Tang M., Wang W. L., 1997, "An Introduction to Workflow Management Systems", National Historical Publications and Records Commission Grant No: 96023, pp.3.

[24] Grimm O., 2004, "A Construction Kit for the Application of Workflow Management Systems in Production Environments", eBRF 2004 Tampere September 20-22 2004, pp.4/15

[25] COMARCH S.A., 25.02.2005, "E-Document and Workflow Management", http://www.comarch.com/en/Solutions/Archiwizacja+Dokumentow+i+Workflow/, 2005

[26] J.Cardoso, R.P. Bostrom, A.Sheth, 2003, "Workflow Management Systems and ERP Systems: Differences, Commonalities and Applications", Journal of Information and Technology Management

[27] Microsoft Corporation, 28.02.2005, "Windows DNA", http://www.microsoft.com/resources/documentation/Windows/2000/server/reskit/enus/Default.asp?url=/resources/documentation/Windows/2000/server/reskit/en-us/iisbook/c06_windows_dna.asp

[28] Extropia The Open Web Technology Company, 28.02.2005, "Introduction to Microsoft DNA", http://www.extropia.com/tutorials/dna/

[29] Redmond, Wash, (February 23, 1999), "Understanding the Microsoft Windows DNA Architecture", http://www.microsoft.com/presspass/features/1999/02-23dna2.asp , 2005(updated)

[30] Matt Publishing Limited, 28.02.2005, "Essentials Microsoft Transaction Server", http://www.dnjonline.com/articles/essentials/iss7_essentials_page2.html, 2001

[31] Rockwell M., Moons J. J., Boutquin P., Brown B., Crouch R., 2000, "Sams Teach Yourself Microsoft Windows DNA Programming in 21 Days", Sams Publishing, pp.24,134

[32] Draft Federal Information, Processing Standards Publication 184, December 21, 1993, the Standard for Integration Definition for Information Modeling (IDEF1X).

[33] Knowledge Based Systems Inc., 01.03.2005, “IDEF1X Data Modeling Method”, http://www.idef.com/IDEF1X.html, 2004(updated)

[34] Essential Strategies Inc., 01.03.2005, “IDEF1X”, http://www.essentialstrategies.com/publications/modeling/idef1x.htm

[35] Alataş B., 2004, “Development of a Web Based Dynamic Scheduling Methodology for a Flexible Manufacturing Cell using Agent Based Distributed Internet Applications”, Master Thesis Middle East Technical University Ankara Turkey.

[36] Watson E. E., Schneider H., Ourso E. J., Rouge B., 1999,”Using ERP System in Education, Communications of the Association for Information Systems”, Vol I, Article 9.

[37] Ricaddi G., 2003, “Database Management With Web Site Development Applications”, Addison Wesley.

[38] Gaydasch A., 1988, “Effective Database Management”, Prentice Hall.

APPENDIX A

USERS MANUAL

The SME DBMS software has been implemented in Integrated Manufacturing Technologies Research Group (IMTRG), Middle East Technical University Mechanical Engineering Department Computer Integrated Manufacturing (METUCIM) laboratory. User, Business, and Data Services of the "SME DBMS" has been mostly written under Visual Basic 6.0. Components for database search and update has been deployed in Microsoft Transaction Server (MTS). The database of the system has been constructed using SQL Server 7.0 for storing global data about the system. Internet Information Server (IIS) has been used to grant access to the web sites ASP and HTML pages, which are designed in Visual InterDev 6.0, a product of Microsoft Visual Studio. The web site of the optimization system is published on Internet Information server at the address: http://cimlabserver.me.metu.edu.tr/SME.

SME Information System is the user interface of the developed software. All of the data of the enterprise can be managed throughout the web pages. The main entrance of the SME Information System is from Login.asp. The users are prompted to type a valid username and password to enter the web site. By that, the identity of the current person is detected and stored, also unattended to the system has been prohibited. An ordinary user will login with Name: *user*, Password: *guest*, he/she can gain the access based on "User". A user can't delete, add, and modify any information contained in the site. If a user has the username and password based on "Engineer", he/she can delete, add, and modify any information

contained in this site. User can log on the system clicking on *Submit* button after indicating valid user name and password or can clear already written user name and password by pressing on *Reset* button. For invalid user name or password entrance, a message written as "*User not found. Check your username and password!"* will come into view.

After logging on to the "SME Information System", user will see the root page in which hyperlinks to various pages, which are classified into seven groups:

1. **Company Data:** The institutional data of the SME is given in this category consisting of *Departments* and *Employees* web pages.

   1.1) Departments: This page gives detailed information about the departments of the SME. According to the selected department name, all the information belonging to the department will be available to the user. *Department_No* is the only property that cannot be modified. The name of the department, the manager and the manager's start date can be updated using modify and submit commands respectively. In order to add a new department click on the *Add* button and after entering the desired data approve by clicking on the *Submit* button. Deletion of a department can be done pressing on *Remove* and *Submit* buttons, respectively after selecting the department by name. Note that add, remove and modify options are only available for "Admin" and "Engineer" roles.

   1.2) Employees: The records containing employee data can be viewed throughout this page. *Employees* page visualize the detailed information about a selected employee according to his/her position and name. *Add/Modify Position* button orientates the user to the Position page enabling to view or update the properties of the selected position or add a new position. After selecting the position, select the employee by name. Modification of employee data except *Person_ID* is possible using Modify and Submit buttons. Addition of a new

employee to the system can be done by clicking on the *Add* and *Submit* buttons respectively. The *Person_ID* value is given automatically by the system. In order to delete an employee click on the *Remove* button and then approve using *Submit* button. *Add*, *Remove*, *Modify* commands are visible only the user is an administrator or engineer.

2. **Commercial Data:** Commercial activity of the SME can be viewed or updated throughout these pages.

2.1) **Companies:** Detailed information about the companies that are commercially related to the SME can be viewed throughout this page. The recorded data about the selected company can be viewed, modified or removed from the system. The Modification of company data is possible by using *Modify* and *Submit* commands respectively. In order to delete a company, after selecting the *Company_ID* click on the *Remove* button and then approve by using *Submit* button. Addition of a new company to the database is also possible. After pressing *Add* button, enter the required data to the data fields and then press *Submit* button. Only the users that have user rights based on "Admin" or "Engineer" can add/remove/delete company data.

2.2) **Sales:** The detailed sales data of the enterprise including the related company, invoice no, sale date etc. is visualized according to the selected sale. Besides the general information about a selected sale, the details about the sold parts and products can be viewed throughout the page. Several products or manufactured parts can be added to a sale. Unit price and the sold quantity of a manufactured part or product can be defined. In order to modify sale's data click on the *Modify* button and after changing the required data fields approve by clicking on the *Submit* button. Addition of a new sale to the system is achieved in two steps. The user switches to the add mode by clicking on the *Add* button.

The following screen consists of two parts. The second part becomes enabled after submitting the data entry of the first part by clicking *Submit New Sale Data* button. Second part contains details about the sold parts and products. Select the part or product from the combo box, enter the required data and click on the save sign. Repeat this action in order to add an other part or product. Press *Submit* button to turn back to the browse mode. *Add, Remove, Modify* commands are visible only the user is an administrator or an engineer.

2.3) Purchases: The search results of the selected purchase are accessed throughout this page. The structure of the Purchases page resembles with the Sales page. The second part contains detailed data about purchased parts, machine tools, raw materials, holders and inserts. And also it is possible to add any other purchased thing except for the categories mentioned above.

3. Assembly/Part Data: Information about the available products, manufactured and purchased parts can be viewed throughout the related pages.

3.1) Products: The general information about the products can be viewed throughout the *Products* page. According to the selected product, the name and number of the components that constitute an assembly are visualized. Also all production dates and the related quantities of the selected product can be accessed by choosing the date from the combo box. In order to modify a product, after clicking on the *Modify* button, change the required data and press *Submit* button. Addition of a new product to the system is achieved in two steps. The user switches to the add mode by clicking on the *Add* button. The following screen consists of two parts. The second part becomes enabled after approving the data entry of the first part by clicking *Submit New Product Data* button. Second part contains details about the products. Enter the production

date and number of produced products at that date then press save sign. In order to add another date and quantity pair press Add New button and repeat the same steps. Also the selected production dates can be removed by clicking on the *Remove* button. To define the components of an assembly, select the manufactured or purchased part from the related combo box, enter the required data and click on the save sign. Press *Submit* button to turn back to the browse mode. Note that *Add, Remove, Modify* commands are visible only the user is an administrator or an engineer.

3.2) **Manufactured Parts:** The properties of the selected manufactured part can be viewed from *Manufactured Parts* page. System gives flexibility to user according to already defined user role for making addition, deletion and modification in the database. Click on *Add* and *Submit* buttons for defining new part in the system. Similar to addition process, *Remove* and *Submit* buttons should be used to remove a part defined by a part id from the database. In order to make necessary updates in the part info, Click on *Modify* and *Submit* buttons, respectively. Note that, for either to define the process plan of new added part or to browse the process plan of an existing part click on *Browse Process Plan* button,. In order to browse or change the part material properties defined in part, press on *Browse Part Material* button. Also the production dates and number of produced parts can be viewed/added/removed/modified for each part.

3.3) **Purchased Parts:** Purchased Part properties are encapsulated in this page. Part name and drawing file name is browsed when a Purchased Part is selected. Click on the drawing file link in order to see the technical drawing of the selected part. Click on *Add* and *Submit* buttons for defining a new purchased part in the system. Similar to addition process, *Remove* and *Submit* buttons should be used to remove a part

defined by a part id from the database. In order to make necessary updates in the part info, Click on *Modify* and *Submit* buttons, respectively.

4. Static Machine Tool/Tool Set/Tool/Part Material/Tool Material Data:

Information about available machine tools, tool sets, tools, part and insert materials used in the system can be viewed and updated throughout the related web pages. Also *Holder Discards* and *Insert Discards* pages are added to the system to follow the holder and insert discards of the SME easily.

4.1) Machine Tool: This page gives detailed information about machine tools. As the machine tool id is chosen, all the information belonging to the machine tool will be available to the user. Correction of any data except machine *Tool_ID* can easily be performed pressing on *Modify* and *Submit* button, respectively. On the other hand, system gives ability to user according to his/her user role to add a new machine tool clicking on *Add* and *Submit* buttons, respectively. Also delete of undesired machine tool can be done pressing on *Remove* and *Submit* buttons, respectively. Note that add, remove and modify of machine tools is only possible for "Admin" and "Engineer" role.

4.2) Tool Set: This page gives link to browse tool sets and related joins to tools. As the tool set id is selected from the combo box, tool set name for the selected tool set and exist status of the available tools defined by tool id will come into view. The state represents the connection between the indicated Tool Set and the Tool. If the state is "Exist" that means the tool was defined in the indicated tool set. If the state is "Not Exist" there is no option that the tool will be used in the specified tool set. When a tool is selected for the tool set the corresponding holder and insert IDs can be viewed on the page. In order to add or delete a tool set click on *Add* and *Submit* or *Remove* and *Submit* buttons, respectively. Note that, whenever a user adds a new tool set, he/she is

responsible to define or add related tools in the indicated tool set by pressing on *Add Tool* button. Removing tool from the indicated tool set can be done by clicking on *Remove Tool* button. Besides, browsing capability of the tool properties will be available pressing on *Browse Tool Properties* button.

4.3) **Tool:** Tool properties can be viewed throughout the Tool page. A holder and an insert constitute a tool. Addition of a new tool to the system means that defining an insert and holder pair that does not exist in the system database with a different *Tool_ID*. While adding a new tool the new *Tool_ID* is given automatically by the system. After selecting the tool type from the combo box select the holder and insert. Only the inserts that have the same type with the defined tool type appears in the combo box. In order to Add/Delete/Modify a tool click on Submit button after clicking on the *Add/Remove/Modify* buttons.

4.4) **Holders:** Holder properties can be viewed or updated throughout this page. While adding a new holder the new *Holder_ID* is given automatically by the system. Use Add/Remove/Modify buttons to switch the related modes and click on the Submit button to approve the action. Note that add, remove and modify buttons are only visible if the user role is "Admin" and "Engineer".

4.5) **Inserts:** This page views the insert properties in a distinctive way. Addition, deletion and modification of an insert is possible clicking on *Add/Remove/Modify* and *Submit* buttons, respectively. Remember that inserts are defined and categorized as either milling or turning insert. So, in the process of defining new tool, user does not have to forget to select the appropriate tool type. Besides that, properties of the insert material, which is indicated for the selected insert, can be viewed clicking on *Browse Insert Material* button.

4.6) **Insert Material:** Information about insert materials is given in this page. Similar to the other pages, addition, deletion and modification action of insert material can be performed pressing on *Add/Remove/Modify* and *Submit* buttons, respectively. *Add/Modify/View Material Constants* button gives link to the Constant Data page to let the user to input/modify or view related constant data values of the material. All the table names and text boxes for table values with relative meanings on the top of them come into sight, when the software takes user to the *Constant Data* page for entering related constant data. Right arrow, at the bottom of the boxes can be used to forward proceed and left arrow can be used to backward proceed through the table values. On the other hand, right end arrow is used to obtain the last record in the recordset, whereas left end arrow indicates the first record in the recordset. User can enter the corresponding value of the indicated table by clicking on the related box and writing the value from keyboard and pressing either left or right arrow for storing new added value. At the top of the page, some parameter ids viewed in the tables and corresponding names are given. Users that have user rights based on administrator or engineer can add new constants or update the existing ones whereas the others can only view them.

4.7) **Part Material:** In this page, user will have a chance to browse part material properties by choosing desired part material id from combo box. Besides that, changing of any data except part material id is possible pressing on *Modify* and *Submit* button, respectively. Removing process of the indicated part material can be done clicking on *Remove* and *Submit* button, respectively. In order to define a new part material, user have to press Add and Submit button, respectively. Note that, *Add/Modify/View Material Constants* button orientates the user to the

Constant Data page to let the user to input/modify or view related constant data values of the material according to their user roles. All the table names and textboxes for table values with relative meanings on the top of them come into sight, when the software takes user to the Constant Data page for entering related constant data, which are showing dependency to part material. Right arrow, at the bottom of the boxes can be used to forward proceed and left arrow can be used to backward proceed through the table values. On the other hand, right end arrow is used to obtain the last record in the recordset, whereas left end arrow indicates the first record in the recordset. User can enter the corresponding value of the indicated table by clicking on the related box and writing the value from keyboard and pressing either left or right arrow for storing new added value. At the top of the page, some parameter ids viewed in the tables and corresponding names are given.

4.8) **Holder Discards:** The data about the holder discards of the SME including the dates and reasons of discards, number of discarded holders for each date are given throughout the page. In order to add a new discard date and related information for a holder, select the holder by Holder_ID and click on the *Add* button in order to switch to the Add mode. After entering the required data press *Submit* button to approve the action. Modification and deletion of a holder discard data can be done by clicking on the Modify/Remove buttons after selecting the holder and the discard date. Click on the Submit button to approve the action.

4.9) **Insert Discards:** The data about the insert discards of the SME including the dates and reasons of discards, number of discarded inserts for each date are given throughout the page. The structure of the page resembles with the *Holder Discards* page.

5. **Static Process Plan/Operational Data:** A detailed information about process plans of available manufactured parts, process plan operations with NC codes, connections between machine tools and these operations, between tool sets and indicated machine tools, tools and related tool sets can be reached using the related hyperlinks.

   5.1) **Process Plan:** As the user selects the part defined with a part id, all the process plans and their properties comes into view. Note that one part defined with a part id may have one or many process plans. Besides that, add, remove and modify procedures of the process plan are same as the procedures operated in the part page. Click on *Browse Operation* button, either to add an operation to new defined process plan or to browse operations of the selected process plan.

   5.2) **Operation:** Operation page gives information about the operations of the process plan. User can browse the operations of the selected part and its process plan choosing the desired sequence no from the combo box. Add, remove and modify procedures can be executed in the same way as in process plan page. Note that a process plan consists of one or many operations, which are associated with one or more machine tool operations. In order to browse machine tool operations of the desired operation, press on Browse Machine Tool Operations button

   5.3) **Machine Tool Operations:** Machine tool operations of the selected operation can be reached from this page. Each entry in machine tool operations represents a physical connection of an operation to one or many workstations, thereby defines alternative process plans. As the indicated part, its alternative process plans and operations are selected from the combo box, assign status of the available workstations to the operations will come into sight. The state represents the connection between the indicated part, process plan and operation sequence no to

the machine tool. If the state is "Assigned" that means the machine tool is a server of the indicated operation. If the state is "Not Assigned" there is no option that the part will be processed in the indicated machine tool. In order to assign the machine tool to the indicated operation, simply click on *Assign* button. On the other hand, user should press on *Release* button to remove the connection between the indicated machine tool and operation. Browsing the machine tool properties is done simply clicking on *Browse Machine Tool* button. In order to view which alternative tool sets may be used in the indicated machine tool, user should press on Browse Tool Set Operations button.

5.4) **Tool Set Operations:** This page shows the alternative tool sets that are planning to be used in the indicated operation. As the indicated part, its alternative process plans, operations and machine tools are selected from their own combo boxes, assign status of the available tool sets to the machine tools will come into view. The state represents the connection between the indicated part, process plan, operation sequence no, machine tool to the tool set. If the state is "Assigned" that means the selected tool set will be used for the indicated operation. If the state is "Not Assigned", there is no option that the tool set will be used in the indicated machine tool. *Assign* button is used to assign the alternative tool set to the indicated machine tool, whereas *Release* button removes the connection between the tool set and machine tool. *Browse Tool Operations* button enables user to browse, add and remove alternative tools defined in the indicated tool sets. Besides that in order to execute browse, add and remove transactions of the tool set, user should click on *Browse Tool Set Properties* button.

5.5) **Tool Operations:** User can get information about alternative tools from this page. When the indicated part, its alternative process plans, operations, machine tools and tool sets are selected from their own

combo boxes, assign status of the available tools to the tool sets will come into view. The state represents the connection between the indicated part, process plan, operation sequence no, machine tool, tool set to the tool. If the state is "Assigned" that means the selected tool can be used for the indicated operation. If the join state is "Not Assigned" there is no option that the tool will be used in the indicated machine tool for the operation. *Assign* and *Release* transaction works similar as in tool set operations page. In order to get information about tools, click on *Browse Tool Properties* button. Links to some related pages are also available at the bottom of the page.

6. **Search:** Search pages enable users to filter the data about the related search category according to the selected parameter.

6.1) **Employee Search:** User can make search for the employees according to six different parameters. In order to search for the employees according to their start dates input the date data to the related fields and click on the *Search* button. The system gives a default interval but the user can change due to the required information. Also using the second search category, the employees whose birth dates are between the dates defined by the user can be filtered by clicking on the Search button after entering the required values. The other search categories search for the employees according to their department, position, education and social security number. In order to perform these searches select the department/position/education /social security number from the related combo box and click on the *Search* button next to the search category. The search results can be seen at the bottom of the search page as a table.

6.2) **Browse Purchases:** Search page for purchases contains six different types of filtering parameters. In order to search for the purchases

according the purchase date after entering the desired date interval to the related fields click on the *Search* button. The system gives a three days time interval as default. The second category gives the opportunity of searching for the purchases according to the payment due date. Use search button after defining the time interval. In order to search for the purchases according to the Suppliers/Invoice no/Purchase ID select the *Supplier/Invoice no/Purchase ID* from the related combo box then click on the *Search* button next to the category. An other search option is to filter the purchases due to the selected purchased item. After selecting the category of the item among *Purchased Part, Holder, Insert, Raw Material, Machine Tool or Other* alternatives, choose the item by ID from the related combo box and click on the *Search* button. The search results are shown as a table at the bottom of the search page.

6.3) **Browse Sales:** Search page for sales work similar to the Browse Purchases page. This page also consists of six search categories. These are according to the *Sale date, Delivery date, Customers, Invoice no, Sale ID and Sold item*. Filtering the sales data according to a selected part or product is possible.

7. Get Help: Hyperlinks contained in this group gives link to help, login, logoff and sitemap screens.

7.1) **Help:** The web site of the SME DBMS is published on http://cimlabserver.me.metu.edu.tr/SME. One can reach detailed explanation about the use of the SME DBMS system at http://cimlabserver.me.metu.edu.tr/SME/Help.htm. The complete web site is also on the "\website\" directory located on the CD. To publish these pages on an IIS server create a new folder on IIS Web "SME" directory for example "C:\Inetpub\wwwRoot\SME" and simply copy all files here.

7.2) **Login:** Login page is the main entrance to the optimization system. Everyone having user name and password can enter the system. Users who don't have user name and password must type "user" as user name and "guest" as password. By default, they gain access rights based on "User". A "User" unable to execute add, delete and modify transactions.

7.3) **Logoff:** Logoff page is the main and safety exit from the system. All sessions parameters will be expired, when the user logoff and must use login page to return the system.

7.4) **Site map:** This page gives the schematic view of the complete web site.

7.5) **FTP Site:** FTP Site link orientates the user to the ftp site to let the user to upload or download specific files.

Note that, hyperlinks available in the root page can be also viewed from left frame of the web site.

APPENDIX B

KEYS&IDS

Manufactured Part, Process Plan, Sequence No, Machine Tool, Tool Set, Tool,Holder, Insert,Purchased Part and Product are related with an ID number, defining its primary key. Database entry ranges of these IDs are given in Table B.1

Table B.1 Database entry ranges

Table	**ID**	**Range**
Manufactured Part	10001: Generic Part ID	10001-19999
Process Plan	20001: Generic Process Plan ID	20001-29999
Sequence No	1: Generic Sequence No	1-999
Machine Tool	30001: Generic Machine Tool ID	30001-39999
Tool Set	40001: Generic Tool Set ID	40001-49999
Tool	50001: Generic Tool ID	50001-59999
Purchased Part	70001: Generic Purchased Part ID	70001-79999
Holder	111: Generic Holder ID	111-999
Insert	11: Generic Insert ID	11-99
Product	3001: Generic Product ID	3001-3999

APPENDIX C

SAMPLE CODES

C.1 The VB code segment to create and destroy an object is as follows:

```
Dim myObject
Dim strSuccess
Dim Rset
 Set myObject=CreateObject("SME.Products")
'the object is created perform some actions using it
' the object has a function called Retrieve (Product_ID as Integer)
strSuccess1=myObject.Retrieve(3001, mRset)
        If mRset1.Recordcount>0 then
        Textbox1.value = trim(mRset1.fields("Product_name").value)
        Textbox2.value = trim(mRset1.fields("Drawing_File_Name").value)
        End If
Set myObject=Nothing
```

In this part of code the *Product* object is created then the name and drawing file name of the product with an ID 3001 are retrieved and the object is destroyed.

C.2 The following code demonstrates the use of the *ObjectContext* and error handling in a program segment:

```
Function AddProduct() as String
Dim mObjectContext as ObjectContext
```

```
Dim mObject as Object

'if an error occurs jump to Errhandler
On Error GoTo Errhandler

'Create the object context
Set mObjectContext=GetObjectContext()
Set mObject=CreateObject("SME.Product")
Call mObject.AddProduct(3001)

'at this instance, another application may have added
'a new product to the database with the same ID
'if so there is already a product available with Product_ID=3001
'if this program tries to add it an error will occur
'if error has not occurred tell the MTS everything is OK

        mObjectContext.SetComplete
        Set mObjectContext=Nothing
        'Report to the programmer that the transaction is complete
        AddProduct="SETCOMPLETE"
        Exit Function
ErrHandler:
        'Roll back all transactions done during the object context is created
        mObjectContext.SetAbort
        Set mObjectContext=Nothing
        AddProduct="SETABORT"
End Function
```

C.3 The following code shows the *Add* method of SME.Employee object.

```
Public Function Add(ByVal Person_ID As Long, Optional ByVal Department_No
As Long, _
Optional ByVal E_Name As String, Optional ByVal Position_ID As Long, _
```

```
Optional Email As String, Optional ByVal Cell_phone As String, _
Optional ByVal Home_phone As String, Optional ByVal Birth_date As Date,_
Optional ByVal Salary As Currency, Optional ByVal Education As String, _
Optional ByVal College As String, Optional ByVal Office As String, _
Optional ByVal Office_phone As String, _
Optional ByVal Address As String, Optional ByVal Start_Date As Date, _
Optional ByVal City As String, Optional ByVal District As String, _
Optional ByVal Social_Sec_No As String, Optional ByVal Social_Sec_Type As
String) As String

On Error GoTo errhandler
Dim Command1 As ADODB.Command

#If usemts Then
   Dim objcontext As ObjectContext
   Set objcontext = GetObjectContext
#End If
Set Command1 = New ADODB.Command
Command1.ActiveConnection = conString
Command1.CommandType = adCmdStoredProc
Command1.CommandText = "spAEMPLOYEE"
Command1.Parameters.Item("@prmPerson_ID") = Person_ID
 Command1.Execute
Set Command1 = Nothing

'Execute the Modify command
Set Command1 = New ADODB.Command
Command1.ActiveConnection = conString
Command1.CommandType = adCmdStoredProc
Command1.CommandText = "spMEMPLOYEE"
Command1.Parameters.Item("@prmPerson_ID") = Person_ID
```

```
Command1.Parameters.Item("@prmDepartment_No") = Department_No
Command1.Parameters.Item("@prmE_Name") = E_Name
Command1.Parameters.Item("@prmPosition_ID") = Position_ID
Command1.Parameters.Item("@prmEmail") = Email
Command1.Parameters.Item("@prmCell_phone") = Cell_phone
Command1.Parameters.Item("@prmHome_phone") = Home_phone
Command1.Parameters.Item("@prmBirth_date") = Birth_date
Command1.Parameters.Item("@prmSalary") = Salary
Command1.Parameters.Item("@prmEducation") = Education
Command1.Parameters.Item("@prmCollege") = College
Command1.Parameters.Item("@prmOffice") = Office
Command1.Parameters.Item("@prmOffice_phone") = Office_phone
Command1.Parameters.Item("@prmAddress") = Address
Command1.Parameters.Item("@prmStart_Date") = Start_Date
Command1.Parameters.Item("@prmCity") = City
Command1.Parameters.Item("@prmDistrict") = District
Command1.Parameters.Item("@prmSocial_Sec_No") = Social_Sec_No
Command1.Parameters.Item("@prmSocial_Sec_Type") = Social_Sec_Type
Command1.Execute
Add = "OK"
#If usemts Then
    objcontext.SetComplete
    Set objcontext = Nothing
    Add = "SETCOMPLETE"
#End If
Exit Function
errhandler:
        #If usemts Then
          objcontext.SetAbort
          Set objcontext = Nothing
         Add = "SETABORT"
```

```
        #End If
 Add = Err.Number & "/" & Err.Source & "/" & Err.Description

Set Command1 = Nothing
End Function
```

C.4 The following code is taken from the Visual InterDev showing the method of creating the *SME.Employee* object for *Add* function throughout the user interface. After the *Add* function is completed the object is destroyed.

```
Function Add()
Dim myObject
dim strSuccess
Set myObject=CreateObject("SME.Employee")
strSuccess=myObject.Add(Textbox1.value,Textbox19.value,Textbox2.value,_
Textbox18.value,Textbox13.value,Textbox11.value,Textbox10.value,_
Textbox7.value, Textbox9.value, Listbox5.getText (), Textbox16.value, _
Textbox14.value, Textbox15.value, Textbox12.value, Textbox8.value, _
Listbox6.getText(),Listbox7.getText(),Textbox3.value,Listbox8.getText())
set myObject=nothing
Response.Write ("Employee added successfully")
End Function
```

C.5 The following SQL statement is required in order to retrieve all production data of a selected product by *Product_ID*

```
CREATE PROCEDURE spRPRODUCT_DATE_By_Product_ID
@prmProduct_ID as int
AS
SELECT *
FROM PRODUCT_DATE
```

WHERE (Product_ID=@prmProduct_ID)
ORDER BY Product_ID
RETURN
GO

C.6 Using this query one can obtain the production dates and the number of products manufactured for the selected product by passing a value to *@prmProduct_ID* parameter.
An example of a stored procedure for data addition is:

CREATE PROCEDURE spAPart
@prmPart_ID as int
AS
INSERT INTO PART(Part_ID)
VALUES (@prmPart_ID)
RETURN
GO

C.7 Data having common Sale_IDs on each table are joined in the following statement:

CREATE VIEW dbo.VIEWProductSale
AS
SELECT dbo.SALES.Sale_ID, dbo.SALES.Company_ID, dbo.SALES.sDate, dbo.SALES.Invoice_no, dbo.SALES.Due_Date,
dbo.PRODUCT_SALES_DETAIL.Product_ID,
dbo.PRODUCT_SALES_DETAIL.Unit_price,
dbo.PRODUCT_SALES_DETAIL.Quantity,
dbo.PRODUCT_SALES_DETAIL.Currency_Type
FROM dbo.PRODUCT_SALES_DETAIL INNER JOIN
dbo.SALES ON dbo.PRODUCT_SALES_DETAIL.Sale_ID = dbo.SALES.Sale_ID

C.8 Database use generally the stored procedures in order to query data. The related stored procedure in SQL Server is invoked as it is shown in the following sample code:

```
Set Command1 = New ADODB.Command
Command1.ActiveConnection = conString
Command1.CommandType = adCmdStoredProc
Command1.CommandText = "spRMACTOOL_PURCH_DETAILS"
Command1.Parameters.Item("@prmPurchase_ID") = Purchase_ID
Command1.Parameters.Item("@prmMacTool_ID") = MacTool_ID

'Recordset properties
Set rsCommand1 = New ADODB.Recordset
rsCommand1.LockType = adLockBatchOptimistic
rsCommand1.CursorLocation = adUseClient
rsCommand1.CursorType = adOpenStatic

'Make the search
rsCommand1.Open Command1, , adOpenStatic, adLockOptimistic
Set outRecordset = rsCommand1.Clone
Set Command1 = Nothing
Retrieve = "OK"
```

C.9 The example code from SME.Milling_Agent :

```
'Getting Tool Data
Dim obj4 As New Tool
mySt = obj4.Retrieve(odti, iRecordset)
odti is the selected Tool_Idfor optimization
For i = 0 To iRecordset.Fields.Count - 1
mTool(i) = iRecordset.Fields.Item(i)
Next i
```

‘Getting Insert_ID of the selected tool
toolii = mTool(3)
‘toolii is the Insert_ID
'Getting Insert Data
Dim obj44 As New Insert
mySt = obj44.Retrieve(toolii, iRecordset)

For i = 0 To iRecordset.Fields.Count - 1
mInsert(i) = iRecordset.Fields.Item(i)
Next i

‘Getting Insert data: tti, ttmi, d, z etc are the attribute values in
Inserts table for the seleted insert
tti = mInsert(0)
ttmi = mInsert(2)
d = mInsert(3)
z = mInsert(4)
landa = mInsert(5)
eax = mInsert(6)
ls = mInsert(7)
da = mInsert(8)
cs = mInsert(9)

C.10 To retrieve all production data of a selected product by *Product_ID* the following SQL statement is required:

CREATE PROCEDURE spRPRODUCT_DATE_By_Product_ID
@prmProduct_ID as int
AS
*SELECT **
FROM PRODUCT_DATE

WHERE (Product_ID=@prmProduct_ID)
ORDER BY Product_ID
RETURN
GO

Using this query one can obtain the production dates and the number of products manufactured for the selected product by passing a value to *@prmProduct_ID* parameter.

An example of a stored procedure for data addition is:

CREATE PROCEDURE spAPart
@prmPart_ID as int
AS
INSERT INTO PART(Part_ID)
VALUES (@prmPart_ID)
RETURN
GO

APPENDIX D

DATABASE TABLES & ATTRIBUTES

Table D.1 Database Tables & Attributes

DB TABLE	ATTRIBUTE	DESCRIPTION
AGV_Agent Properties of the AGVs in the System	AGV_ID	Key attribute defining the selected AGV
	Type	AGVType
	Brand	Brand of AGV
	Status	Status
	Config_String	Configuration String
	Comp_Name	Controller Computer Name
Assembly_Parts Manufactured Parts of an assembly	Product_ID	ID of the Product (assembly)
	Part_ID	Manufactured parts of the assembly
Assembly_ Pparts Purchased Parts of an assembly	Product_ID	Produt defining the assembly
	Ppart_ID	Purchased parts of the assembly
Banks	Bank_Name	Bank names
Buffer_Agent Properties of the Buffer Agents	Buffer_ID	Key attribute defining the buffer
	Type	Buffer Type

Table D.1 Database Tables & Attributes (continued)

	Brand	Brand of Buffer
	Status	Status
	Config_String	Configuration String
	Comp_Name	Controller Computer Name
Cell_Agent Properties of the Cell Agents in the system	Cell_ID	Cell ID
	Type	Cell Type
	Description	Description of the cell
	Status	Status
Cell_AGV AGV Agents of a Cell	Cell_ID	Cell ID
	AGV_ID	AGV ID
Cell_WS Work Stations of a Cell	Cell_ID	Cell ID
	WS_ID	Work Station ID
City List of cities in Turkey	City_Code	City
	City	City Name
Companies Supplier and Customer Companies' data	Company_ID	Company ID
	Company_Name	Name of the Company
	Country	Country
	Address	Address
	City	City
	District	District
	Email	Email address
	Zip_code	Postal code
	web_site	Web site of the company
	Tax_no	Tax number
	Tax_office	Tax office

Table D.1 Database Tables & Attributes (continued)

	Bank_name	Bank name
	Branch_no	Branch no defining the bank branch
	Branch	Branch name
	Contact_Person	Contact person from the company
	Contact_Phone	Phone number of the contact person
	Conatct_Mail	Email address of the contact person
Con_Cor_Bh Optimization constants used in calculations	Insert_Mat_ID	Insert Material ID
	Part_Mat_ID	Part Material ID
	Dubi	Correlation constant for optimization
	Pq	Correlation constant for optimization
Con_Cor_Bm Constants used in tool life equations for milling	Insert_Mat_ID	Insert Material ID
	Part_Mat_ID	Part Material ID
	Bm	Correction coefficient in tool life formula for milling
Con_Cor_Bp Constants used in tool life equations for milling	Insert_Mat_ID	Insert Material ID
	Part_Mat_ID	Part Material ID
	Part_Mat_Sca_ID	Part Material Scale ID (exists :1 / not exists :2)
	Part_Mat_Pro_Type_ID	Part Material Production Type
	BhNMax, BhNMin	Max. and Min. values of correction coefficient in tool life formula for milling

Table D.1 Database Tables & Attributes (continued)

	Bp	Correction coefficient in tool life formula for milling
Con_Cor_Kh Constants used in optimization calculations	Sigma Min	Min. tensile strength (MPa)
	SigmaMax	Max. tensile strength (MPa)
	Kh	Correction coefficient for cutting force and cutting power for ultimate tensile strength of steels in face milling.
Con_Cor_Kx Constants used in optimization calculations	CsMin	Minimum side cutting edge angle (deg)
	CsMax	Maximum side cutting edge angle (deg)
	Kx	Correction factor in cutting force and cutting power equations for cutting edge angle in face milling
Con_Cut_Force Constants used in cutting force equations	Mil_Type_ID	Milling type ID
	Insert_Mat_ID	Insert Material ID
	Part_Mat_ID	Part Material ID
	Czp	Constant in cutting force equation for milling
	Ez	Exponent in cutting force equation for milling
	Uz	Exponent in cutting force equation for milling

Table D.1 Database Tables & Attributes (continued)

	Nz	Exponent in cutting force equation for milling
	Rz	Exponent in cutting force equation for milling
	Bz	Exponent in cutting force equation for milling
Con_Sel_Fz Constants used in optimization calculations	Part_Mat_ID	Part Material ID
	RaMin	Minimum center-line-average roughness (mm)
	RaMax	Maximum center-line-average roughness (mm)
	Mil_Pre_ID	Milling Presicion ID
	Fz	Feed per tooth in milling (mm/tooth)
Con_Tool_Life Constants used in optimization calculations	Mil_Type_ID	Milling Type ID
	Insert_Mat_ID	Insert Material ID
	Part_Mat_ID	Part Material ID
	BMin	Minimun milling width (mm)
	BMax	Maximum milling width (mm)
	AMin	Maximum permissible depth of cut for turning (mm)
	AMax	Minimum permissible depth of cut for turning (mm)

Table D.1 Database Tables & Attributes (continued)

	FzMin	Minimum feed per tooth in milling (mm/tooth)
	FzMax	Maximum feed per tooth in milling (mm/tooth)
	Cv	Constant in tool life formula for milling or cutting speed constant in British units (ft/min)
	M	Exponent in tool life formula for milling or the slope of the objective function curve
	Bv	Exponent in tool life equation for milling
	Ev	Exponent in tool life formula for milling
	Uv	Exponent in the tool life formula for milling
	Rv	Exponent in tool life formula for milling
	Nv	Exponent in tool life formula for milling
Con_Tur_Mac Constants used in optimization calculations	Part_Mat_ID	Part Material ID
	Insert_Mat_ID	Insert Material ID
	TCv	Constant in tool life formula for turning or cutting speed constant in British units (ft/min)
	TN	Spindle speed (Rpm)

Table D.1 Database Tables & Attributes (continued)

	TG	Slenderness ratio of the chip in turning or absolute value of the gradient vector
	TY	Exponent of the chip cross sectional area in turning
	TCp	Cost of machine preparation cost per component (TL) or cutting force constant in British units for turning (psi)
	TGp	Exponent of the slenderness ratio in cutting force equations in turning
	T1_Zp	Exponent of the chip cross sectional area in actual
Country Country List	Country	Country Name
Currency_Types Currency Types	Currency_Type	Currency types
Departments Departments of the SME	Department_No	Department numbers of the SME
	Department_Name	Department Name
	Person_ID	Manager of the Department

Table D.1 Database Tables & Attributes (continued)

	Manager_Start	Start date of the department manager
District Districts of each city in Turkey	City_Code	The City code of the District
	District_Code	Code defining the District
	District	District Name
Employee Employees' data	Person_ID	Person ID, unique for each employee
	Department_No	Department No
	E_Name	Name of the employee
	Email	Email address
	Position_ID	Position ID of the employee
	Address	Address
	City	City
	District	District
	Cell_phone	Cell phone
	Home_phone	Home phone
	Birth_date	Birth date
	Salary	Salary
	Education	Education (Bs. Ms. etc.)
	College	Graduated university
	Office	Office
	Office_phone	Office phone
	Start_Date	Start date of the employee
	Social_Sec_No	Social Security Number

Table D.1 Database Tables & Attributes (continued)

	Social_Sec_Type	Social Security Type
Fax_Numbers Fax numbers of the commercially related companies	Fax_Number	Fax number of the related company
	Company_ID	Company ID
Holder Available Holders	Holder_ID	Holder ID
	Holder_Name	Holder Name
Holder_Discards Records of discarded holders	Holder_ID	ID of the discarded Holder
	DDate	Discard Date
	Quantity	Discarded Quantity
	Explanation	Reasoning of the discard
Holder_Purch_Details Details of holder purchases	Purchase_ID	Purchase ID unique for each purchase
	Holder_ID	Holder ID
	Unit_Price	Purchasing price of an holder
	Currency_Type	Currency type of the purchasing price
	Quantity	Number of purchased holders
Input_Data Input Data for optimization calculations	Optimization_ID	Optimization ID
	Tool_ID	ID of the Tool used in the operation
	Tool_Set_ID	ID of the Tool Set used in the operation

Table D.1 Database Tables & Attributes (continued)

	MacTool_ID	ID of the Machine Tool used in the operation
	Sequence_No	Sequence number
	Process_Plan_ID	Process Plan ID
	Part_ID	Part ID
	Cost_Over	Labor & Overhead Cost
	Cost_Tool	Tool Cost
	Cost_Jig_Fix	Jig, Fixture Cost
	Wid_Cut	Width of cut
	Dep_Cut	Depth of cut
	Sur_Rough	Surface Roughness
	Dia_Tol_Cut	Diametral Tolerance
	Ave_Rou_Rough	Average Roughness of Roughing
	Ave_Rou_Finish	Average Roughness of Finishing
	Tdepth_Div	Division for total depth of cut
	Mil_Type_ID	Milling Type ID
	Mil_Pre_ID	Milling Precision ID
	Mount_Type_ID	Workpiece Mounting Type ID
	Time_UnLo	Loading, Unloading Time of one Workpiece
	Time_Set	Setup time of Jigs and Machine Tool
	Time_Chan	Tool Changing Time
	Time_Proc	Process Adjusting and Quick Return Time
	Lot_Size	Lot Size
	Grind_Com	Number of Tool Grinding Components

Table D.1 Database Tables & Attributes (continued)

	Mil_Direc_ID	Milling direction ID (i.e Up Milling ID=1 etc.)
Insert_Discards Records of discarded inserts	Insert_ID	ID of the Discarded Insert
	DDate	Discard Date
	Quantity	Discarded Quantity
	Explanation	Reasoning of the discard
Insert_Purch_Details Details of insert purchase details	Purchase_ID	Purchase ID
	Insert_ID	ID of the Purchased Insert
	Unit_Price	Purchasing price of an Insert
	Currency_Type	Currency type of the purchasing price
	Quantity	Number of the purchased inserts
Inserts Available inserts	Insert_ID	Insert ID
	Insert_Name	Insert Name
	Insert_Mat_Name	Insert Material ID
	Insert_Diameter	Diameter of the Insert
	Num_Insert_Teeth	Teeth number
	Face_Cle_Ang	Face Clearance Angle
	Arbor_Elas	Modulus of elasticity of Arbor (Gpa)
	Arbor_Leng	Arbor Length (mm)
	Arbor_Dia	Arbor Diameter (mm)
	Side_Cut_Edge_Ang	Side Cutting Angle
	End_Cut_Ang	End Cutting Angle
	Nose_Rad	Nose Radius

Table D.1 Database Tables & Attributes (continued)

	Insert_Type	Insert Type (i.e. Milling or Turning Insert)
List_Error Accured Errors within the system	Error_ID	Error ID
	Error_Description	Description of the Error
	Error_Help_File	Error Help File Name
	Error_Date_Time	Date and Time of the Error
	Error_Source	Source of the Error
	Error_Status	Error Status
List_Heat_Trea Heat Treatment definitions for part materials	Part_Mat_Heat_Trea_ID	Part Material Heat Treatment ID
	Part_Mat_Heat_Trea_Name	Part Material Heat Treatment Name
List_Insert_Mat_Name Insert Material definitions	Insert_Mat_ID	Insert Material ID
	Insert_Mat_Name	Insert Material ID
List_MacTool_Type Machine Tool Type definitions	MacTool_Type_ID	Machine Tool Type ID
	MacTool_Type_Name	Name of the Machine Tool (Milling or Turning)
List_Mil_Direc Milling direction definitions	Mil_Direc_ID	Milling Direction ID
	Mil_Direc_Name	Milling Direction Name

Table D.1 Database Tables & Attributes (continued)

List_Mil_Mat_Sca Part Material Scale definitions	Part_Mat_Sca_ID	Part Material Scale ID
	Part_Mat_Sca_Name	Part Material Scale Name
List_Mil_Pre Milling Precisions (very accurate, accurate etc.)	Mil_Pre_ID	Milling Precision ID
	Mil_Pre_Name	Milling Precision Name
List_Mil_Type Milling Type definitions	Mil_Type_ID	Milling Type ID
	Mil_Type_Name	Milling Type Name
List_Optimization_Cri Optimization Criteria definitions	Optimization_Criteria_Type_ID	Optimization Criteria Type ID
	Optimization_ Criteria_Type_Name	Optimization Criteria Type Name
List_Optimization_Pass Pass types of the operations (Multipass turning, single pass millng and turning)	Optimization_Pass_Type_ID	Optimization Pass Type ID
	Optimization_Pass_Type_Name	Optimization Pass Type Name
	Dep_Cut_Pass	Depth of cut
	Rec_Tool_Life	Recommended Tool Life
List_Optimization_Tech Used optimization techniques	Optimization_Tech_ID	Optimization Technique ID
	Optimization_Tech_Name	Optimization Technique Name
List_Pro_Type	Part_Mat_Pro_Type_ID	Part Material Production Type ID

Table D.1 Database Tables & Attributes (continued)

Production types of the part materials	Part_Mat_Pro_Type_Name	Part Material Production Type Name
Machine_Tool Available Machine Tools	MacTool_ID	Machine Tool ID
	MacTool_Name	Machine Tool Name
	MacTool_Type_ID	Machine Tool Type ID
	Motor_Pow	Motor Power (KW)
	Tare_Pow	Tare or Idle Power (KW)
	Mech_Eff	Mechanical Efficiency
	Num_Spin	Total Number of spindle speeds
	Num_Feed	Total Number of feed rates
	Feed_Val_Que	Feed Rate Values in mm/min
	Speed_Val_Que	Spindle Speed Values in Rpm
Machine_Tool_Operations Operation data after specifying the Machine Tool	MacTool_ID	Machine Tool ID
	Sequence_No	Sequence Number
	Process_Plan_ID	Process Plan ID
	Part_ID	Part ID
MacTool_Purch_Details Purchase details of machine tools	Purchase_ID	Purchase ID
	MacTool_ID	ID of the purchased Machine Tool
	Unit_Price	Purchasing price of the Machine Tool
	Currency_Type	Currency type of the purchasing price

Table D.1 Database Tables & Attributes (continued)

Material Part Material data	Part_Mat_ID	Part Material ID
	Part_Mat_Name	Part Material Name
	Part_Mat_Strength	Part Material Strength
	Part_Mat_Hardness	Part Material Hardness
	Part_E	Part Material Elastic Module
	Part_Mat_Heat_Trea_ID	Part Material Heat Treatment ID
	Part_Mat_Pro_Type_ID	Part Material Production Type ID
	Part_Mat_Sca_ID	Part Material Scale ID
Operations Operation data	Sequence_No	Sequence No
	Process_Plan_ID	Process Plan ID
	Part_ID	Part ID
	Sequence_Name	Sequence Name (Milling or Turning)
	G_Code_File	File name containing G Codes of the operation
	Drawing_File_Name	Drawing file of the part
	Estimated_Time	Estimated operation time
	Len_Cut	Length of cut
	Ini_Dia_Cut	Initial diameter of the part
	Fin_Dia_Cut	Final diameter of the part
	Person_ID	Person ID of the Operator
Optimization Optimization data definitions	Optimization_ID	Optimization ID
	Tool_ID	ID of the Tool used in the operation

Table D.1 Database Tables & Attributes (continued)

	Tool_Set_ID	ID of the Tool Set used in the operation
	MacTool_ID	ID of the Machine Tool used in the operation
	Sequence_No	Sequence No
	Process_Plan_ID	Process Plan ID
	Part_ID	Part ID
	Optimization_Criteria_Type_ID	Optimization Criteria Type ID
	Optimization_Pass_Type_ID	Optimization Pass Type ID
	Optimization_Tech_ID	Optimization Technique ID
	Optimization_Description	Description
	Optimization_Date	Optimization Date
Other_Purch_Details Purchase details of all sort of items defined by the user except the specified purchase categories	Purchase_ID	Purchase ID
	O_Name	Name of Purchased Item
	Unit_Price	Unit price of the purchased item
	Currency_Type	Currency type of the purchasing price
	Quantity	Purchased quantity
	Unit	Unit (kg,m,etc.)
Output_Data Optimization results	Optimization_ID	Optimization ID
	Tool_ID	ID of the Tool used in the operation
	Tool_Set_ID	ID of the Tool Set used in the operation
	MacTool_ID	ID of the Machine Tool used in the operation

Table D.1 Database Tables & Attributes (continued)

	Sequence_No	Sequence No
	Process_Plan_ID	Process Plan ID
	Part_ID	Part ID
	Pass_No	Pass No
	Analysis_Type	Optimization Analysis Method
	Cost	Optimum Cost
	Time	Optimum Time
	Feed_Rate	Feed Rate
	Spindle_Speed	Spindle Speed
	Feed_PerTeeth	Feed per Teeth
Part Manufactured Parts' data	Part_ID	Part ID, unique for each part
	Part_Name	Part Name
	Part_Mat_ID	Part Material ID
	Part_Length	Part Length
	Revision	Revision
	Rev_Date	Revision Date
	Drawing_File_Name	Drawing File of the Part
	Raw_Length	Raw Length of the Part
Part_Date Production dates and quantities of a Part	Part_ID	Part ID
	Production_Date	Production Date of the Part
	Quantity	Number of Manufactured Parts
Part_Sales_Detail Sales details of the manufactured parts	Sale_ID	Sale ID,unique for each Sale
	Part_ID	ID of the sold Part
	Unit_Price	Selling Price of a Part

Table D.1 Database Tables & Attributes (continued)

	Currency_Type	Currency type of the selling price
	Quantity	Number of sold Parts
Phone_Numbers Phone numbers of commercially related companies	Phone_Number	Phone Number of the Company
	Company_ID	Company ID
PLRD_Agent	PLRD_ID	PLRD ID
	Type	Type of the PLRD
	Brand	Brand of the PLRD
	Status	Status
	Config_String	Configuration String
	Comp_Name	Computer Name
Position	Position_ID	Position ID
	Position_Name	Position Name
PParts_Purch_Details	Purchase_ID	Purchase ID
	PPart_ID	Purchased Part ID
	Unit_Price	Purchasing Price of the Part
	Currency_Type	Currency Type of the Purchasing Price
	Quantity	Number of Purchased Parts
Process_Plan Process Plan data	Process_Plan_ID	Process Plan
	Part_ID	Part ID
	Revison	Revised or not
	Rev_Date	Date of Revision
	Type	Type of Revision
	Description	Description

Table D.1 Database Tables & Attributes (continued)

Product_Date Production dates and quantities of an Assembly	Product_ID	Assembly ID
	Production_Date	Production Date
	Quantity	Number of produced Products
Product_Sales_Detail Sales details of the products	Sale_ID	Sale ID
	Product_ID	ID of the sold Product
	Unit_Price	Selling Price of a Product
	Currency_Type	Currency Type of the selling price
	Quantity	Number of sold Products
Products Assembly data	Product_ID	Product ID (Assembly ID)
	Product_Name	Product Name
	Drawing_File_Name	Name of the File containing assembly drawings
Purchased_Parts Purchased Parts' data	PPart_ID	Purchased Part ID
	PPart_Name	Purchased Part Name
	Drawing_File_Name	Drawing File Name of the Purchased Part
Purchases General purchase data of the SME	Purchase_ID	Purchase ID, unique for each Purchase
	Company_ID	Supplier Company
	pDate	Date of Purchase
	Invoice_no	Invoice Number
	Payment_term	Payment Term
	Payment_type	Payment Type

Table D.1 Database Tables & Attributes (continued)

	Due_Date	Payment Due Date
RawM_Purch_Details Purchase details of the Part Materials	Purchase_ID	Purchase ID
	Part_Mat_ID	Part Material ID
	Unit_Price	Purchasing Price of the Part Material
	Currency_Type	Currency Type of the Purchasing Price
	Quantity	Purchased Quantity
	Unit	Unit (m, kg, etc.)
Robot_Agent Properties of the Robots in the System	Robot_ID	Robot ID
	Type	Type of the Robot
	Brand	Brand of the Robot
	Config_String	Configuration String
	Comp_Name	Controller Computer Name
Robot_Buffer	Robot_ID	Robot ID
	Buffer_ID	Buffer ID
Robot_PLRD	Robot_ID	Robot ID
	PLRD_ID	PLRD ID
Sales General Sales data of the SME	Sale_ID	Sale ID, unique for each Sale
	Company_ID	Customer Company ID
	sDate	Date of Sale
	Invoice_no	Invoice Number
	Delivery_address	Delivery Address
	Payment_type	Payment Type
	Payment_term	Payment Type
	Due_Date	Delivery Due Date

Table D.1 Database Tables & Attributes (continued)

Sensitivity Optimization Sensitivity analysis parameters' data	Param_ID	Analysis Parameter ID
	Param_Value	Analysis Parameter Value
Sensitivity_Graph Sensitivity Graph input data	Optimization_ID	Optimization ID
	Param_ID	Analysis Parameter ID
	Option_ID	Drawing Option
Social_Sec_Types Social Security Types	Social_Sec_Type	Social Security Type
Task_List Tasks list performed by the Agents	Work_Order_No	Work Order No
	Sequence_No	Sequence No
	Task_ID	Task ID
	Task_Customer_Agent_ID	
	Task_Server_Agent_ID	
	Bidding_Start_Time	
	Bidding_End_Time	
	Queue_Wait_Start_Time	
	Queue_Wait_End_Time	
	Own_Task_Start_Time	
	Own_Task_End_Time	
	Task_Start_Time	
	Task_End_Time	
	Task_Status	
Tasks Tasks' input data	Task_ID	Task ID
	Task_Parameter	Task Parameter
	Task_Description	Task Decription
	Estimated_Time	Estimaated Time
Tool	Tool_ID	Tool ID
	Holder_ID	A Tool is defined by an

Table D.1 Database Tables & Attributes (continued)

Available Tools	Insert_ID	Insert and Holder pair
	Tool_Type	Tool Type (Milling or Turning)
Tool_Operations Operation data after specifying the used Tools	Part_ID	Part ID
	Process_Plan_ID	Process Plan ID
	Sequence_No	Sequence No
	MacTool_ID	ID of the Machine Tool used in the Operaiton
	Tool_Set_ID	ID of the Tool Set used in the Operaiton
	Tool_ID	ID of the Tool used in the Operaiton
Tool_Set Defined Tool Sets	Tool_Set_ID	Tool Set ID
	Tool_Set_Name	Tool Set Name
Tool_Set_Item Tools that constitute a Tool Set	Tool_Set_ID	Tool Set ID
	Tool_ID	Tool Sets consists of Tools
Tool_Set_Operations Operation data after specifying the used Tool Set	Part_ID	Part ID
	Process_Plan_ID	Process Plan ID
	Sequence_No	Sequence No
	MacTool_ID	ID of the Machine Tool used in the Operaiton
	Tool_Set_ID	ID of the Tool Set used in the Operaiton
Users	User_Name	User Name

Table D.1 Database Tables & Attributes (continued)

Users of the DBMS	User_Password	User Password
	User_Description	User Description
	User_Roles	User Role (Admin, Engineer or User)
Work_Order Work Order input data	Work_Order_No	Work Order No
	Part_ID	Part ID
	Process_Plan_ID	Process Plan ID
	Issue_Date	Issue Date
	Completion_Time	Completion time
	Due_Date	Due Date
	Order_Status	Status of the order
	Priority	Priority
Work_Order_Operations Work Order Operations data	Work_Order_No	Work Order No
	Sequence_No	Sequence No
	Start_Time	Operation Start Time
	End_Time	Operation End Time
WS_Agent Properties of the Work Stations in the system	WS_ID	Work Station ID
	Type	Type of the Work Station
	Brand	Brand of the Work Station
	Status	Status of the Work Station
	Config_String	Configuration String
	Comp_Name	Controller Computer Name

Table D.1 Database Tables & Attributes (continued)

WS_Robot Definition of which Robot serves to which Work Station	WS_ID	Work Station ID
	Robot_ID	Robot ID

APPENDIX E

MTS COMPONENTS OF THE DBMS

Figure E.1 and E.2 shows the MTS screens containing the system DLLs.

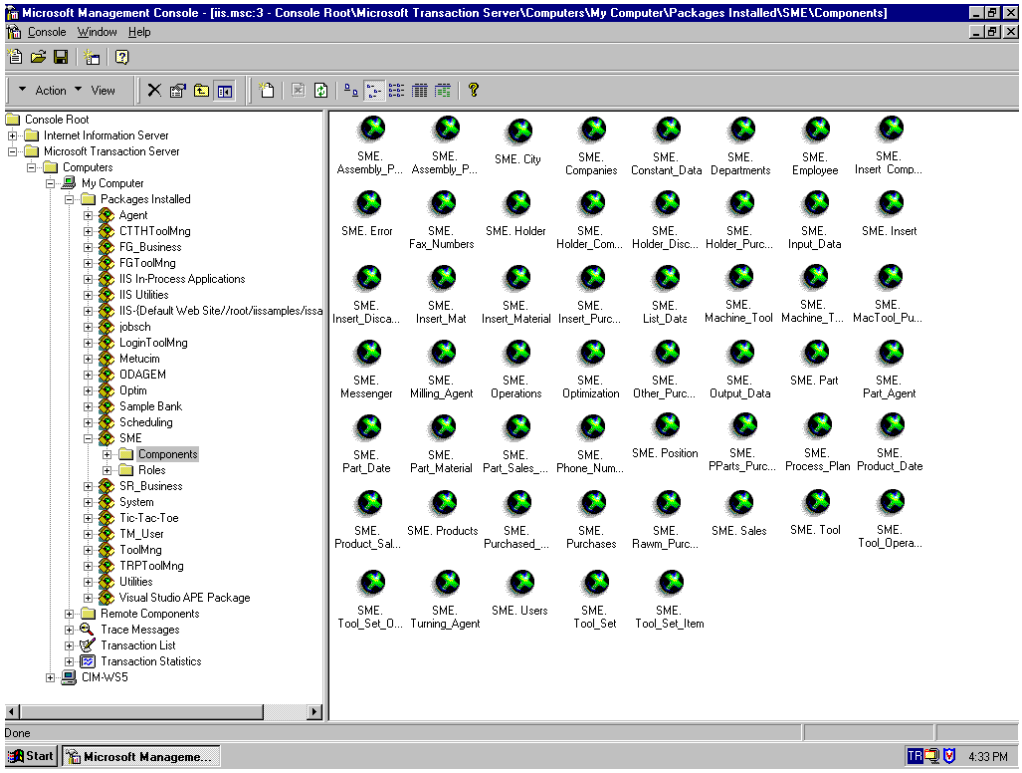
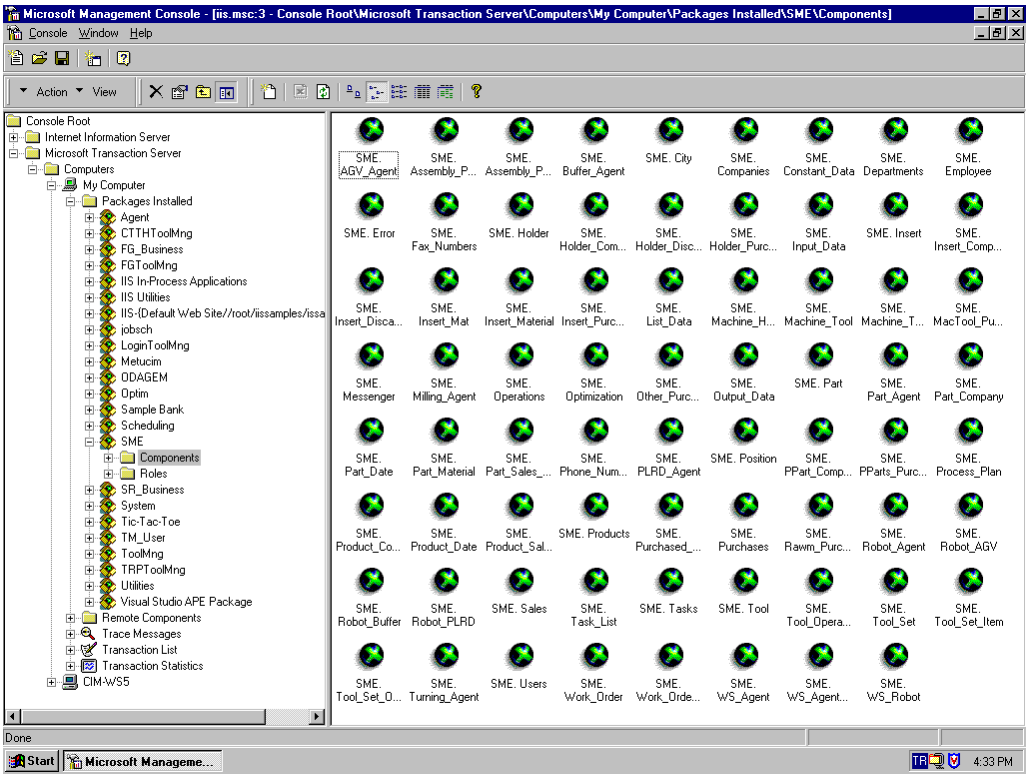


Figure D.1 MTS Screen containing database objects of "Optim" and "SME Information System"



FigureD.2 MTS Components of "SME Information System", "Optim" and "Agent"