

DESIGNING AN INFORMATION SYSTEM
FOR MATERIAL MANAGEMENT
IN ENGINEER-TO-ORDER ORGANIZATIONS

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

BY

ERDOĞAN DEDE

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

DECEMBER 2006

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

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

Prof. Dr. Çağlar Güven
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.

Assoc. Prof. Dr. Tayyar Şen
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Canan Sepil	(METU, IE)	_____
Assoc. Prof. Dr. Tayyar Şen	(METU, IE)	_____
Assoc. Prof. Dr. Levent Kandiller	(METU, IE)	_____
Asst. Prof. Dr. Seçil Şavaşaneril	(METU, IE)	_____
Önder Okyay (M.S.)	(TÜBİTAK-SAGE)	_____

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

Name, Last name : Erdoğan, Dede

Signature :

ABSTRACT

DESIGNING AN INFORMATION SYSTEM FOR MATERIAL MANAGEMENT IN ENGINEER-TO-ORDER ORGANIZATIONS

Dede, Erdoğan

M.S., Department of Industrial Engineering

Supervisor : Assoc. Prof. Dr. Tayyar Şen

December 2006, 94 Pages

In this thesis, an information system is designed and developed for engineer-to-order organizations to improve the traditional Bill-of-Material by handling variants of products and components efficiently. A database is developed to store the related information about inventories and configuration management in an effective way. The improved Bill-of-Material provides a common structure to access stored information for material management purposes. A model, based on network, is presented and included into the system for calculating time required to produce components and to make subassemblies or assemblies with the current inventory levels. The system is applied to TÜBİTAK-SAGE, which is an engineer-to-order organization carrying out Research and Development projects for Defense Industry.

Keywords: Information System, Material Management, Prototype Development, Bill-of-Material, Engineer-to-Order.

ÖZ

MÜHENDİSLİK FİRMALARINDA MALZEME YÖNETİMİ İÇİN BİLİŞİM SİSTEMİ TASARIMI

Dede, Erdoğan

Yüksek Lisans, Endüstri Mühendisliği Bölümü

Tez Yöneticisi : Doç. Dr. Tayyar Şen

Aralık 2006, 94 Sayfa

Bu tezde, siparişe-özel-tasarım yapan organizasyonlar için geleneksel ürün ağaçları geliştirilerek ürün ve malzeme çeşitliğini etkin kontrol edecek bilişim sistemi tasarlanmıştır. Envanter ve konfigürasyon yönetimi ile ilgili bilgilerin etkin kaydedilmesi için bir veritabanı tasarlanmıştır. Geliştirilen ürün ağacı kaydedilen bilgilere ürün yönetimi amacıyla ulaşılması için ortak bir yapı sağlar. Ürün ve ara-ürünlerin güncel envanter seviyeleri kullanılarak, şebeke temelli, bir model sunulmuş ve tasarlanan bilişim sisteminde bu model kullanılarak ürünlerin ve ara-ürünlerin monte edilmesi için gereken sürenin hesaplanmasında kullanılmıştır. Bilişim sistemi Savunma Sanayi'nde Araştırma ve Geliştirme projeleri yapan bir siparişe-özel-tasarım organizasyonuna, TÜBİTAK-SAGE'ye, uygulanmıştır.

Anahtar Kelimeler: Bilişim Sistemi, Malzeme Yönetimi, Prototip Geliştirme, Malzeme Listesi, Siparişe-Özel-Tasarım.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my supervisor Assoc. Prof. Dr. Tayyar Şen for his guidance, advice, criticism, encouragements and insight throughout the research.

I would also like to thank Önder Okyay and Beril Boyacıoğlu for their suggestions and comments.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	v
ACKNOWLEDGMENTS.....	vi
TABLE OF CONTENTS	vii
LIST OF TABLES.....	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xii
CHAPTER	
1. INTRODUCTION	1
2. LITERATURE SURVEY	4
3. SYSTEM ANALYSIS AND PROBLEM DEFINITION.....	14
3.1 Information about TÜBİTAK-SAGE.....	14
3.1.1 Structures in the Institute.....	15
3.1.2 General Information about Ongoing Projects.....	17
3.1.3 System Development Process.....	18
3.1.4 Components Production	20
3.1.5 In-House Production Capabilities.....	23
3.1.6 Configuration Management.....	24
3.2 Problem Definition.....	27
4. PROPOSED SYSTEM	32
4.1 Basic Assumptions	33
4.2 Model-Base of the Proposed System	34
4.2.1 Documentation Module.....	34
4.2.2 Storage Module.....	35
4.2.3 Inventory Module	35
4.1.4 The Product Tree Module.....	37
4.3 Database.....	39
4.3.1 Database Tables	40

4.3.2 Normalization	46
4.4 User Interface Design	47
4.5 Benefits of the Proposed System (Verification)	50
5. THE PLANNING MODEL	51
5.1 The Product Structure	51
5.2 The Network	57
6. CONCLUSION	64
REFERENCES	67
APPENDICES	
A. PROCESS LIST FORM	70
B. QUALITY CONTROL FORM	72
C. INVENTORY LIST FORM	75
D. KNAL FORM	77
E. RELATIONSHIP DIAGRAM	79
F. USER INTERFACES OF THE DEVELOPED SYSTEM	81

LIST OF TABLES

TABLES

Table 1 General Numerical Information.....	18
Table 2 Class of Items According to Quality Control	21
Table 3 Number of Distinct Components	28
Table 4 Simple Product Structure.....	51
Table 5 Component List with Lead Times.....	54
Table 6 Process List with Process Times.....	54
Table 7 Available WIP CIs	55
Table 8 Component Requirement for CIs	55
Table 9 Assembly Operations	56
Table 10 CIs Quantities.....	58
Table 11 Component Quantities.....	59
Table 12 Component Nodes Duration	59
Table 13 Durations of Process Nodes for CIs	61
Table 14 Durations of Assembly Nodes for CIs	61
Table 15 Configuration Item Precedence Relation.....	62
Table 16 CPM/PERT Solution to CI 1	63

LIST OF FIGURES

FIGURES

Figure 1 Multi-level Generic BOM of Ballpoint Pen Assembly with generic components	5
Figure 2 Relationship between different BOMs.....	6
Figure 3 Simple BOM for Typist Chair.....	7
Figure 4 Generating Conventional BOM.....	7
Figure 5 BOMO for Souvenir Clock Manufacturing	8
Figure 6 Virtual BOM Example for Mountain Bike	10
Figure 7 Variations in Raw Material Decomposition.....	11
Figure 8 Sample Process Based Multi-Level BOM in Textile Industry.....	13
Figure 9 Organizational Chart of TÜBİTAK-SAGE.....	16
Figure 10 A Sample Project Structure	17
Figure 11 Integrated BOM System Modules	32
Figure 12 Cycling Relations.....	39
Figure 13 Entity Relationship Diagram	41
Figure 14 General Flowchart for the Designed System.....	48
Figure 15 BOM for the Simple Product.....	52
Figure 16 Production of the Final Configuration Item	53
Figure 17 Production Process of the Configuration Item 1.....	57
Figure 18 Duration of Process Node for CI 2	60
Figure 19 Network Diagram for Production Process	62
Figure 20 Sample Process List	71
Figure 21 Sample Quality Control Form (Page 1)	73
Figure 22 Sample Quality Control Form (Page 2)	74
Figure 23 Sample Inventory List	76
Figure 24 Sample KNAL	78
Figure 25 Relationship Diagram of the Designed Database	80
Figure 26 The Main Menu	81

Figure 27 The Main Page of the Documentation Module	81
Figure 28 Document Add.....	82
Figure 29 Document Changes	82
Figure 30 The Main Menu of the Inventory Module.....	83
Figure 31 Defining Inventory Item Properties	83
Figure 32 The Page for Entering Items to Stores	84
Figure 33 Items Available List.....	85
Figure 34 Use Item Operations.....	85
Figure 35 Items Used List	86
Figure 36 The Main Menu of the Storage Module.....	87
Figure 37 Store Definition	87
Figure 38 Cell Lists.....	88
Figure 39 Store Cell Definition	89
Figure 40 The Main Menu of the Product Tree Module.....	89
Figure 41 The Page for Modifying Product Trees.....	90
Figure 42 The Page for Defining CIs.....	90
Figure 43 Link Document Page.....	91
Figure 44 Link List of a CI.....	91
Figure 45 Entering Required Quantity for a CI.....	92
Figure 46 Material Requirements List	93
Figure 47 Entering Quantity For Planning BOM.....	93
Figure 48 Time Planning for a CI production	94
Figure 49 Add Variation Page	94

LIST OF ABBREVIATIONS

BOM	: Bill-of-Materials
BOMO	: Bill-of-Materials and Operations
BOMfr	: Bill-of-Manufacture
CAM	: Computer Aided Manufacturing
CI	: Configuration Item
CPM	: Critical Path Method
CNC	: Computer Numerical Control
EDM	: Electric Discharge Machining
ERP	: Enterprise Resource Planning
GBOM	: Generic Bill-of-Materials
KNAL	: Given Configuration Number List
MRP	: Material Requirements Planning
MRP II	: Manufacturing Resource Planning
PERT	: Project Evaluation and Review Technique
R&D	: Research and Development
VPC	: Virtual Product Constructor
WIP	: Work in Process

CHAPTER 1

INTRODUCTION

Nowadays, more products based on high technology are being developed in defense industry. The development of these products requires the production of various prototypes consisting of continuously evolving components. Engineering-to-order organizations are specialized on development of this kind of products. These organizations need special information systems, different from the ones in make-to-stock organizations, to support material management in production.

The main features of Defense Industry are as follows (TÜBİTAK, 2006):

- Government is the single customer.
- High technology must be applied.
- Products must be safe, reliable and information about them is to be kept secret.
- The organizations in Defense Industry are required to be strong, big and safe.
- Dependency to foreign countries is required to be at the lowest level.

TÜBİTAK-SAGE (The Scientific & Technological Research Council of Turkey-Defense Industries Research and Development Institute) is a non-profit, technology producing, engineer-to-order institute. The primary objective of the organization is to make research and development (R&D) projects for Defense Industry. The output of these projects is highly customer specified, high valued products or prototypes. The institute is involved in 20-25 ongoing projects on average at any time. A typical project continues as long as five years and can require up to 90-100 professionals (mostly engineers), on average, during this duration.

In order to develop products or prototypes, components are designed first and tests are performed in specified conditions. Required components for tests are produced in small quantities by selected suppliers and delivered to the institute. While

components are being produced outside, the design of these components might be changed or modified completely, as the nature of R&D activities. This type of production causes several problems: (1) Numerous lists have to be kept. (2) It is hard to find correct components prior to tests. (3) Files in different formats are prepared for storing information. (4) Time planning for production takes long time. (5) Data are kept separately between divisions and data flow is manual and insufficient. (6) Current ERP packages cannot be used effectively.

Production and material management in the institute is different from make-to-stock organizations enormously. While make-to-stock organizations produce high number of undiversified items, the institute has small number of distinct items produced outside and these items evolve continuously. This type of production needs a material management system different than the one for classical make-to-stock organizations.

In this thesis, in order to integrate data stored in different formats and places, first a relational database management system using Microsoft Access is designed to store required information. Second a product tree structure is used as a main system to access this information for production monitoring. The system is composed of four modules and product variants can be created and managed easily. The system is prepared for simultaneous access from all users. Then, a network problem is formulated for production planning. The required time for assembling a product or sub-product is found by utilizing the CPM/PERT method.

The second chapter gives literature on Bill of Materials, since a multi-level BOM structure is given as a basis for the proposed system on material management.

In the third chapter, general information about the organization and projects of TÜBİTAK-SAGE is given to evaluate the complexity and volume of the work performed. The role of configuration management in the system development process is given. The development and production of prototypes in the current system is described and the problems of this process are given in detail.

The fourth chapter incorporates the design information system for proposed material management system. The assumptions for the designed system are given and the designed database structure is introduced. In addition, user interface of the proposed system is presented.

The fifth chapter explains the planning model by means of an example. The network created for material management and solution to this network are presented in this chapter.

In the sixth chapter, the work done is summarized. The main advantages and drawbacks of the newly developed material management system are discussed. Suggestions for future work related to the subject matter are also presented.

CHAPTER 2

LITERATURE SURVEY

Engineer-to-order organizations produce custom made or one-of-a-kind products (Bertrand and Muntslag, 1993). This type of products need numerous prototype productions before final products are delivered to customers. In these organizations, prototypes and their components are not fixed, and they are maturing over time. Therefore, there exist a great number of variants for prototypes and component in the system. The traditional Bill-of-Material is not appropriate for prototype production since the traditional BOM has fixed structures with standard components. When the traditional BOM structure is used in engineer-to-order organizations, there would be one BOM for each variant of components and prototypes. In that case, the number of BOM reaches to huge numbers. The literature on production in engineer-to-order organizations and different types of BOM systems are searched to find how variants of components, products and operations are handled without increasing the number of BOMs.

To handle product variants, generic Bill-of-Materials (GBOM) concept is introduced by Veen and Wortman (1987) especially for assemble-to-order environments. In assemble-to-order environments, end-items mostly have a number of features where each feature can have several options. Due to different combinations of options, the number of end-items can be enormous. Therefore, a manufacturing BOM for every single end-item cannot be defined. GBOM describes a range of similar products implicitly with a single embracing BOM and it is composed of generic items which describe a range of items. When a customer orders a specific product in the product range, the matching BOM is generated from GBOM by substituting generic items with the chosen specific items. A GBOM for a ballpoint pen is given in Figure 1 (Veen and Wortman, 1987). Since some combinations of parameter values might define a non-marketable product, a configuration control function is added to the

generic BOM concept so that a specific BOM can only be generated for a marketable product. GBOM can describe a range of product by generically, but the structure and options for features must be known when GBOM is defining contrary to the case in engineer-to-order organizations.

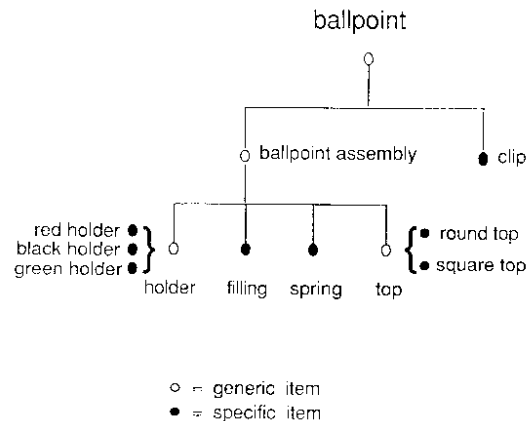


Figure 1 Multi-level Generic BOM of Ballpoint Pen Assembly with generic components

A closely related study to this thesis is given by Bertrand and Muntslag (1993) in that main differences between engineer-to-order manufacturing and the make-to-stock manufacturing (which was the basis for the development of MRP II software) are discussed. In their work, important characteristics of the engineer-to-order manufacturing are given as the important role of the customer order, customer-specific product specifications, and the product and production uncertainty. These characteristics differ substantially from the basic assumptions of MRP II, which are the production of standard products with a well-known BOM and product routing in advance. Therefore, it is argued that engineer-to-order environments ask for a completely different production control system. Hence, a production control framework for engineer-to-order production was proposed. Within this framework, three BOM structures are defined for engineer-to-order environment: (1) Standard BOM, (2) Historical BOM and (3) Project BOM. Project BOMs are prepared with the use of standard BOMs. Standard BOMs are product functions which were

previously accepted as product standards within the engineering and design department. During product construction, historic BOMs of previous projects are used with standard BOMs to prepare project specific BOMs. The relationship between these BOMs is illustrated in Figure 2 (Bertrand and Muntslag, 1993). However, engineer-to-order organization differs in terms of product development processes. In this thesis, a BOM structure for engineer-to-order organizations which develop different prototypes for each customer order is considered. In this type of organizations, there does not exist standard BOM widely accepted for all prototypes since each prototype is produced for completely different products, but there are numerous historic BOMs and a single project BOM for each project and historic BOMs are created from project BOMs.

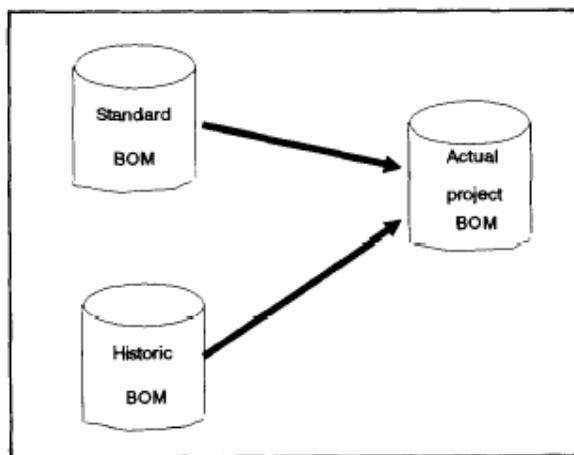


Figure 2 Relationship between different BOMs

A generic BOM system based on Programming Language Notation for both product and component variant handling is proposed by Olsen et al. (1995). A simple BOM for typist chair and the flow for generating conventional BOM for this specific item are given in Figures 2 and 3, respectively (Olsen et al., 1995). The generic structure of a product is constructed from generic components which can have alternative specifications. The alternative specifications for components are included in the body of generic component definition. When a specific variant for a product is constructed

by traversing generic BOM structure, a specific value among generic component specifications is chosen for each one of generic components included. The defined constraints in goes-into-relations in generic structure become active when specification list is prepared. The specifications complying with these constraints are listed for the user when a product variant is constructed. The proposed generic BOM system offers a high level of flexibility to describe not only variants but also constraints between components in goes-into-relations of BOM structure, but the system takes a well known product structure with well known alternative components and therefore it is more appropriate for make-to-stock productions.

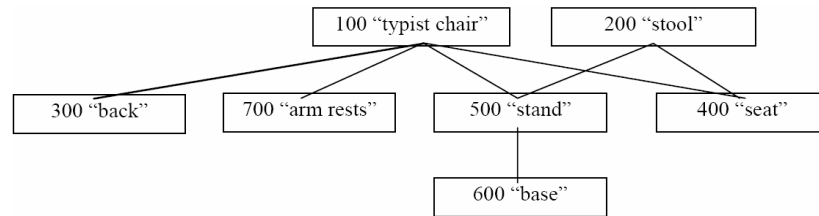


Figure 3 Simple BOM for Typist Chair

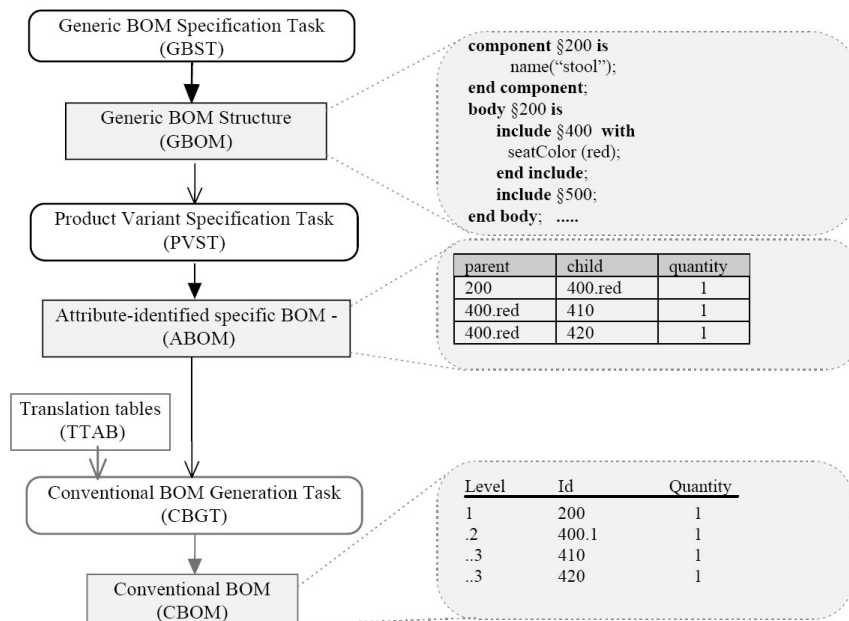


Figure 4 Generating Conventional BOM

Hastings and Yeh (1992) proposed a concept of bill of manufacture (BOMfr) which combines routings with traditional BOMs to provide material requirements data. Tatsiopoulous (1996) discussed the reason for separation of routing data entities from BOM data entities and gave the consequences of unifying BOM and routings on the basic functions of production planning and control. Jiao et al. (2000) proposed a data structure called generic Bill of Materials and Operations (BOMO) by unifying routings and BOM to handle variants resulting from both product changes and processes. Generic BOMO has four elements which are generic product, generic goes-into relationships, generic operation and generic planning. In the paper, an implementation of the generic BOMO in customized souvenir clock manufacturing is given. The sample BOMO for souvenir clock manufacturing is given in Figure 4 (Jiao et al., 2000). Contrary to engineer-to-order organizations, all possible processes and components are assumed to be known when generic BOMO is defining.

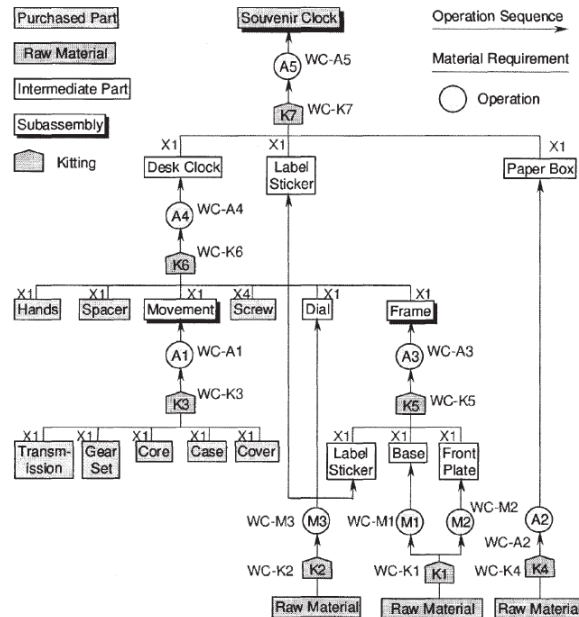


Figure 5 BOMO for Souvenir Clock Manufacturing

A related study to engineer-to-order organization is given by Little et al. (2000). It is discussed that there are four product characteristics in engineer-to-order organiza-

tions: (1) Super-value products are produced in an uncertain and complex environment. (2) Products are designed and developed innovatively to customers for fitness. (3) Products are not standard. (4) Customers change their requirements over time of the manufacture of the product. However, variant handling is not discussed in their work. Since lead times are high for this type of production, planning and scheduling reference model for engineer-to-order organizations is given to assist company management in a review of their planning and scheduling processes. The important elements for integrated planning and scheduling in an engineering-to-order system are given as product configuration, master production scheduling, design planning, project requirements planning, shop floor scheduling, and assembly scheduling.

One of the closely related studies to this thesis is given by Olsen et al. (2001) in that a Visual Product Constructor (VPC) is proposed for engineer-to-order environments which need a different type of manufacturing planning and control systems than serial production manufacturing. It is argued that the traditional BOM structure is effective for serial production where products have the same product structure, but in engineer-to-order environments each customer order has its own specifications, and it is not possible for these environments to maintain static product structures. In that paper, variant handling in engineer-to-order manufacturing is compared with variants handling in serial production manufacturing; variants in serial production manufacturing are usually represented by a generic BOM in which all possible variations have to be known when the generic BOM is created, but each new order can describe a new variant in engineer-to-order environments and the complete product structure cannot be known until design phase is completed. Therefore, it is concluded that a generic BOM structure cannot be maintained for engineer-to-order environments. Project oriented activities like construction of buildings, bridges, ships and offshore installations are given for engineer-to-order production cases in which the products may be too different to be described as variants. Therefore, engineer-to-order manufacturers describe products only by drawings and single-level part lists. VPC is a visual BOM representing the information structure for a product and it provides a structure to store and retrieve diverse information like contracts,

specification, calculations, drawings, part descriptions. The emphasis of the tool is more on information other than automation. A structure for a new product or order can be constructed in a very short time by a cut & paste process from historical BOM in VPC. This initial BOM structure is modified in design and engineering phases by modifying component attributes, adding new components and deleting unnecessary components. The visual BOM for mountain bike example is given in Figure 6. When the tool is designing, it is assumed that there exists certain similarities between products and therefore the tool is not suited for a situation where each order does not have any similarities. However, the study in this thesis is for engineer-to-order organization in which each order can have no similarities, therefore all methodology of VPC cannot be used in this thesis.

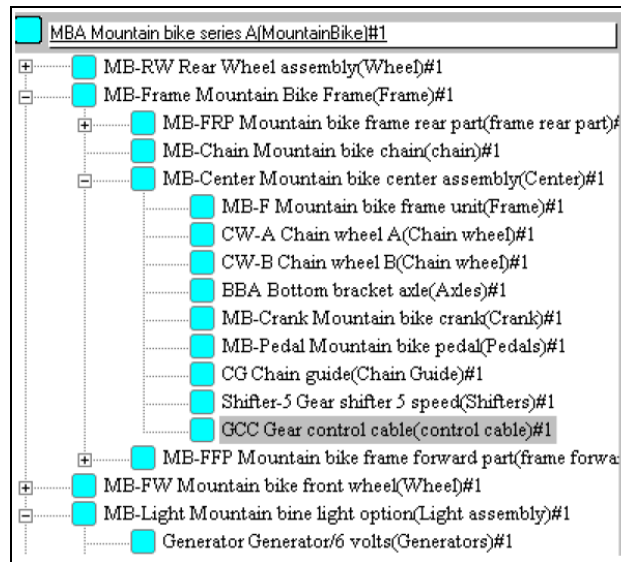


Figure 6 Virtual BOM Example for Mountain Bike

Another study related to variant handling was done by Vegetti et al. (2002) for process industry. The variant handling in process industry is different than engineer-to-order organizations and it is similar to the case in make-to-stock organizations with one difference; products can be produced by decomposing of raw materials. An example for variants in decomposition is given in Figure 7 (Vegetti et al., 2002).

However, three drawbacks of traditional BOM structure is given for variant handling which are also applicable for engineer-to-order organizations: (1) In the traditional BOM where each variant is considered as a separate product and the structure of each variant is stored which leads to a dramatic increase in physical storage requirements. (2) Storing each variant as a BOM may also create logical data inconsistency if data changes associated with a specific component is not simultaneously made in all BOM structures in which the component participates. (3) Previously proposed BOM structures consider variants only at the finished product level, leaving aside variants of intermediate assembly products. Complex BOM was proposed to overcome these drawbacks. In the proposed system, each variant structure is derived when it is required in order to reduce stored data volume. The model minimizes the links between products and their variants, in this way it enables changes in product structure to be carried out without affecting the associated variants. The complex BOM structure can represent products with decomposition of raw materials in the petroleum and meat industries. The traditional BOM structure can only represent product by composition of raw materials, components or subassemblies. The model in this paper is not utilized since the product variants are assumed to be known before BOM declaration.

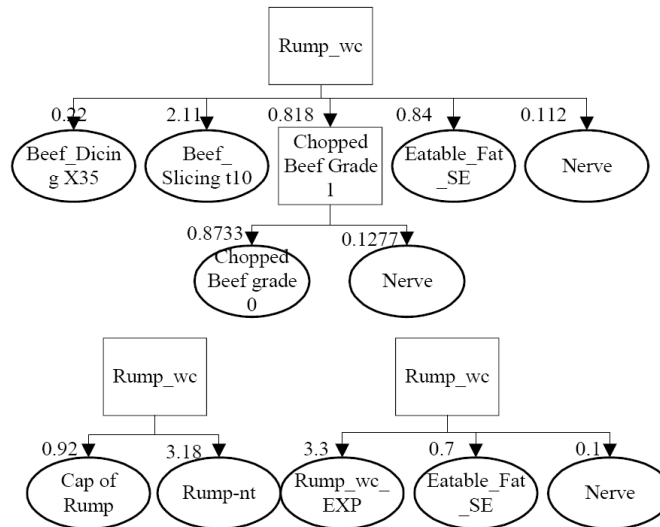


Figure 7 Variations in Raw Material Decomposition

Aydın and Güngör (2003) presented an effective approach to handle excessive product variation problem for mass customization in textile industry. The given approach handles product information in a relational database environment for multi-product and multi-process production systems. The approach aims to reduce the efforts when defining products to the production systems for generating BOM and executing material requirements planning (MRP). A multi-level BOM representation of sample product in textile industry is given in the left section of Figure 8 (Aydın and Güngör, 2003). In the given approach, BOMs are considered as multi-level where each level of a BOM matches to a process. The process based representation of sample product A is given in the middle section of Figure 8. Some consecutive processes are merged into one process. After merging consecutive processes, the resulting BOM structure is given in the right section of Figure 8. In textile industry, most of the variants of products are developed upon changing just one or two components in one process or changing consumptions of components. Therefore, components and consumption are represented generically in the proposed approach. However, the BOM structure can be changed in engineer-to-order organizations. The proposed system takes all product definitions at the order receiving process regardless of the product status (i.e. an existing product, a variant of an existing product or a new product). However, BOMs are changing continuously until products are finalized in engineer-to-order organizations.

The literature is also searched to store BOM tree structure. Guoli et al. (2003) proposed a tree structure storage model to record a product tree. The proposed model uses two tables to record parent components and child components. Each component or parent item location in product tree is given as a combination of three fields, which product tree the object belongs to, on which level in the tree it lays, and of which position on the level it has been fixed. Child component table has two special link fields. The first one indicates the parent level of the current component and the second one indicates whether a child of the current item exists. The study in this paper is so general that the storage model does not handle variants in engineer-to-order organizations.

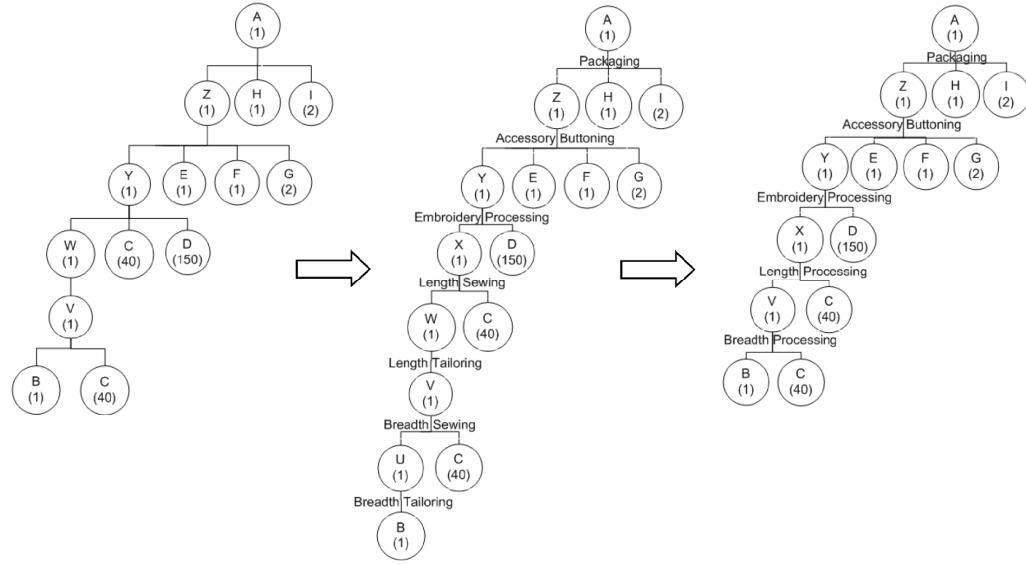


Figure 8 Sample Process Based Multi-Level BOM in Textile Industry

The generic structure given previously represents varieties only for components or subassemblies. However, Du et al. (2003) discussed that inclusion of product varieties also differentiates assembly routing processes, therefore inclusion of process variations to generic product structure is needed. In this paper, variety synchronization for assembly-to-order production is discussed. The generic model for product and process variants includes (1) generic components, (2) generic assembly operations and (3) generic planning for synchronization of product and process variants. The paper reported a case study conducted in a refrigerator company to demonstrate the feasibility and potential of the proposed approach.

In the literature, variant handling is generally proposed for make-to-stock environments where product structures and all alternative components are assumed to be known when BOM is defining contrary to the case in engineer-to-order organizations. Two closely related studies for variant handling in engineer-to-order organizations are found in the literature. These two studies proposed standard BOM, historic BOM and project BOM for engineer-to-order organization. However it is assumed that there are some commonalities between each order in the latter studies.

CHAPTER 3

SYSTEM ANALYSIS AND PROBLEM DEFINITION

System analysis is a study to determine the system structure and the requirements of TÜBİTAK-SAGE for material management in prototype development along with system development process. In this thesis, only material management in system development process is given for an engineer-to-order organization to develop prototypes. Problems related to prototype development are given only in terms of material management perspective.

3.1 Information about TÜBİTAK-SAGE

TÜBİTAK-SAGE is a R&D institute, which works in the area of defense industry since its foundation in 1972. The institute performs primarily R&D work, as well as testing and consultancy. The main working area of TÜBİTAK-SAGE is the design and development of weapon systems and munitions.

The institute functions in a project-oriented manner. A project is started right after the project is approved by the Administrative Board of the institute. Various development projects, regarding the defense industry, are being executed in the institute. These projects can be divided into two groups: Customer based projects and in-house projects. In customer based projects, the requirements are set by the customer via technical specification documents. The primary customer of TÜBİTAK-SAGE is the Ministry of Defense. Other important customers are defense industry companies such as ASELSAN, ROKETSAN, TAI, etc. and institutes of TÜBİTAK. The requirements of in-house projects are determined by the institute considering the promising areas of technology which will be needed in the following projects. While customer based projects are financed by customers, in-house projects are funded by the institute's own resources.

3.1.1 Structures in the Institute

TÜBİTAK-SAGE has a matrix organizational structure which is composed of functional and project oriented structures. Functional groups are formed according to their technical expertise. The functional organizational chart of TÜBİTAK-SAGE is given in Figure 9. There are mainly three levels in the given organizational chart. These levels are departments, groups and divisions. There are two departments which are composed of various groups. The groups are composed of different divisions.

Besides the functional organization given above, project teams are formed by assigning different professionals from various departments for each project. A sample project structure is given in Figure 10. A project manager is assigned to each project by the director of the institute. Each project is composed of several work packages. For the bigger projects, a group of work packages are put together under a so called “sub-project”. Since the activities performed in one project are completely different than activities in the other projects, the project structure, the work-packages and the sub-projects are completely different between projects.

Leaders of sub-projects and work packages are determined by project managers and associated functional managers. There are also administrative and technical staffs (system engineers, configuration manager, project management representative, quality assurance representative, etc.) reporting directly to the project manager. Owing to the matrix organizational structure and multi-project environment of TÜBİTAK-SAGE, it is generally the case that project managers, work package leaders and other project members may work in other projects.

The functional organization structure is static, but a change can be proposed by the director of the institute. After, the proposal is approved by the Science Committee of TÜBİTAK, the new functional organization becomes active. The project structure is only static during a certain phase of a project. After a phase is completed, new work packages, sub-projects are formed. If a change is required during a phase, the change is proposed by project manager, and it is approved by managers of the departments.

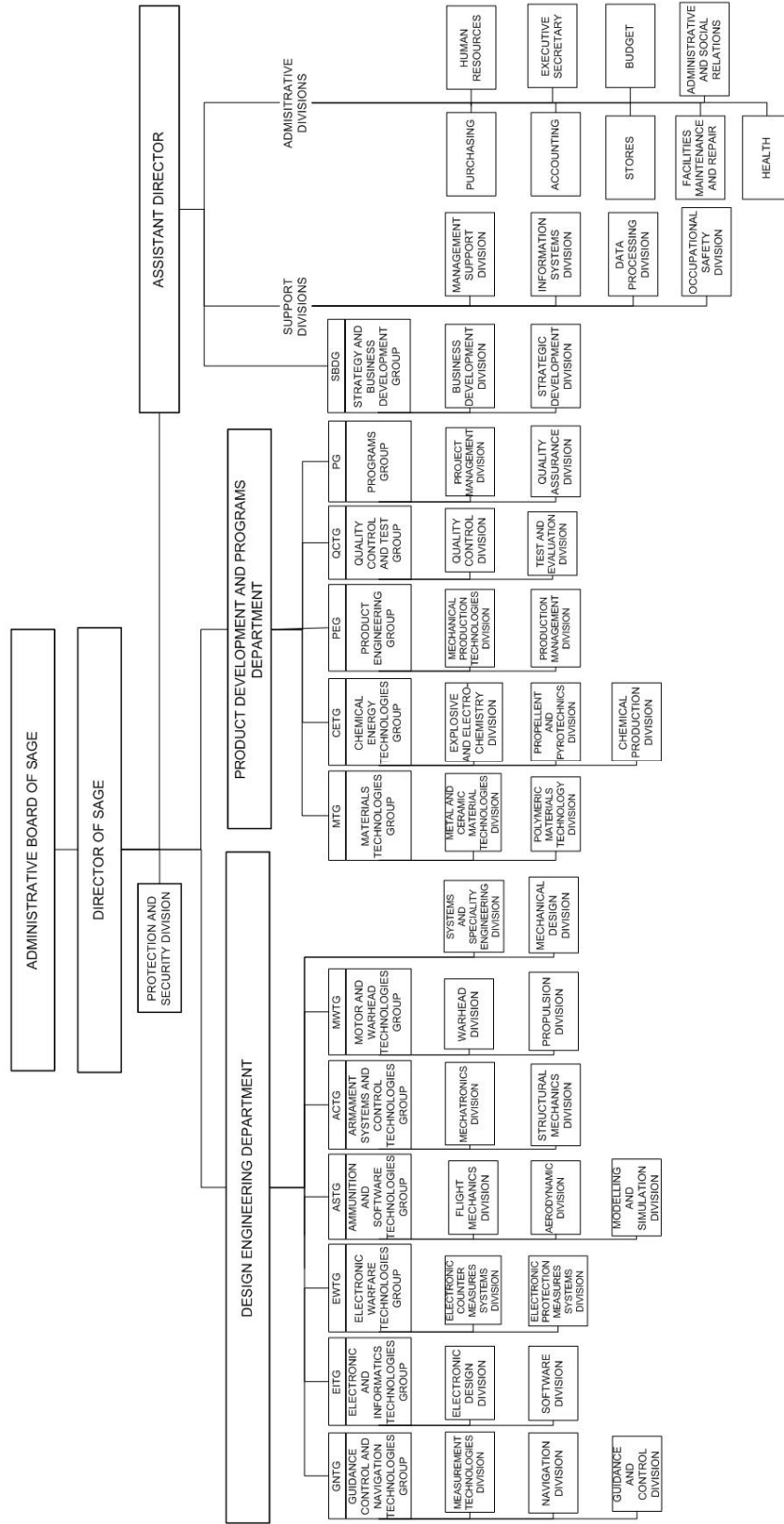


Figure 9 Organizational Chart of TÜBİTAK-SAGE

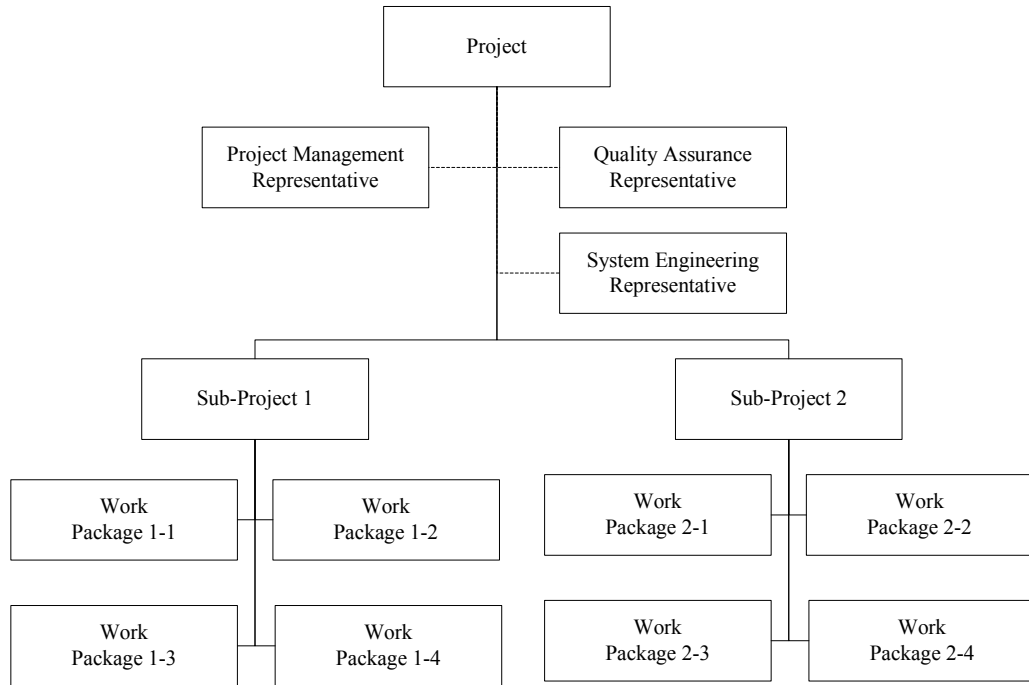


Figure 10 A Sample Project Structure

3.1.2 General Information about Ongoing Projects

Numerical information about the projects of the institute for the last seven years is given in Table 1. Currently 25 projects (14 customer based, 11 in-house projects) are active in the institute. Generally the durations of the projects are 2 to 5 years. The financial value of the active projects in 2000 is taken as the unit value, and the financial values of the active projects of other years are normalized using this unit value. The total financial value of active projects in 2000 is greater than one million dollars.

It can be seen from Table 1 that there exist a continuous increase in the financial value of the projects. The size of the projects can be considered to be directly proportional with the project financial value. As the project size increases, the workforce requirement and the complexity also increases. This complexity brings complex products having greater number of distinct components.

Table 1 General Numerical Information

Years	2000	2001	2002	2003	2004	2005	2006
Number of in-house Projects	12	19	8	8	11	13	11
Number of Customer Based Projects	13	17	12	8	6	12	17
Total Number of Active Projects	25	36	20	16	17	25	28
Contracted New Projects	7	8	4	3	4	7	10
Number of New Products	2	2	1	3	0	4	2
Project Financial Values	1.00	11.11	11.08	10.84	14.98	20.02	75.13
Number of Workers	174	177	204	241	266	274	301
Number of R&D Engineers	85	87	110	141	160	165	192

3.1.3 System Development Process

The system development process is the evaluation of a particular new system from the time when a need for it is recognized and a feasible technical approach is identified, through its development and introduction into operational use. A typical system development has the following characteristics (Alexander and Sweet, 2003).

- Requires complex effort
- Meets an important user need
- Requires several years to be completed
- Made up many interrelated tasks
- Involves several different disciplines
- Performed by several organization
- Has Specific budget and schedule

Since development of a new system has great risks, system development process is divided into phases in which technology is defined and matured into viable concepts, which are subsequently developed and readied for production. In TÜBİTAK-SAGE, system development process is divided into four phases which are given timely arranged order as conceptual design, preliminary-design, detailed design and production line qualification phases. Material management begins in conceptual design phase and finishes at the end of production line qualification phase in a project.

In the conceptual design phase, system performance requirements down to the subsystem level are identified, potential system concepts fulfilling those requirements are determined and, functional and physical characteristics of the new system are defined finally. In this phase there is no or minimum level of production activities.

In the preliminary design phase, some numerical models and formulas are prepared. At the same time, some preliminary prototypes are produced and tests are performed with these prototypes. The results taken from these studies are reviewed. If the requirements are covered by the results, then the preliminary design phase is completed, else preliminary design phase restarted.

In the detailed design phase, design gained from previous phase reaches to a state of completeness. Tests with prototypes continue in this phase. Analyses, simulations, designs are detailed in this phase. After ground tests with prototypes are completed, flight tests are started. In these tests, performance of the prototypes is measured with respect to military standards.

Production line qualification phase is included in some of the projects of TÜBİTAK-SAGE. This phase is for the validation of the production capability of the facility designed and/or developed specifically for the needs of the project. The personnel, materials, items, specific purpose tools, manufacturing processes; test, storage and transportation capabilities required for production of the end product are validated to

be adequate and appropriate. A qualified and validated production line is prepared in this phase.

In each phase of the system development life cycle, various items such as system/subsystem specifications, design reports, technical drawings, instructions, softwares, tables, engineering and production documents are prepared. In these phases, various prototypes and components are produced by utilizing these documents. Especially, towards the end of system development life cycle, a great number of developed systems and their components are needed for validation tests. Besides, in the production line qualification phase, numerous prototypes are produced to validate the production line and the end products so that both end product and the production line have desired quality. Furthermore, it is necessary to deliver definite number of prototypes and their subsystems to customers, if sample prototype deliveries to customers are specified in the project contract. Generally, the systems and subsystems mentioned above are produced by selected suppliers, while a limited number of components are produced in-house.

3.1.4 Components Production

In the “Preliminary Design” and “Detailed Design” phases of projects, components, subsystems or systems are designed by project staff working in Design Engineering Department, using tools such as AutoCAD, Unigraphics, Mechanical Desktop or similar design or analysis software packages. Due to the nature of the projects carried out in TÜBİTAK-SAGE, these components, subsystems and systems are nonstandard, one-of-a-kind and complex.

The designs made by design team are then passed to project members working in the Product Engineering Department which is responsible for the manufacturing of the designed components. These components are generally of mechanical, electronic, chemical, composite, pyrotechnic, etc. type. Generally, the components are manufactured by selected suppliers, then subassemblies and final assembly is

performed in the institute. Data of processes performed on a component is stored in MS Excel documents. A sample process list form is given in Appendix A.

The components delivered by the suppliers are subjected to quality control by the Quality Control Division. Components are measured and controlled according to technical documents (drawings, technical specifications, source control specifications, etc.). All measurements are recorded in quality control forms and reported to the associated project personnel. A Quality Control Form is given in the Appendix B. Components are classified into seven different groups in terms of quality. These are given in Table 2.

Table 2 Class of Items According to Quality Control

Accepted	Components complying with technical documents. This class of item can be used in assembly and tests.
Improper	Components with some or all measurements are out of bound. This kind of items cannot be used unless they are further evaluated and labeled as “Conditionally Accepted”.
Conditionally Accepted	Components with some measurements are out of boundary but the rest of measurements are in boundary. Usage of this kind of components is permitted under defined circumstances.
Scrap	This class of items cannot be used in tests and productions
Valuable Scrap	Although this class of components is scrap, they may be used for specific purposes. Therefore they need to be stored.
Trial	Components in this class are the ones produced for testing purposes in the initial phases of projects.
No Need to Quality Control	These components are standard parts and can be used in test or production.

The component is labeled as “Accepted” if all measurements are found to be in the tolerance band defined in the technical document. This class of components can be used in any test and production. If at least one measurement is not in the given range for a specific component, the component is classified as “Improper”. This class of

items are detached from the production batch and placed in a special area prepared for improper components, in order to be evaluated in detail.

A board, which is named as “Material Review Board”, is constructed to evaluate improper components. The members of the board are the person who makes quality control of the component, the project manager, the sub-project manager and/or the work package leader of the related project. The board classifies improper components further to be used in production and tests. The board can decide that some improper components can be used in productions and tests under some circumstances. These components are labeled as “Conditionally Accepted”. This type of components is evaluated again before usage in terms of performance for a test or production. To decide the effects of improperness on component performance, related work-package leaders, sub-project managers and project manager come together and decide whether the component performance is supposed to be affected by the improperness. If it is decided that the improperness has no severe effects on components performance, those components can be used in tests.

Material Review Board may also decide that an improper component cannot be used in any tests under any circumstances; these items are labeled as “Scrap”.

Standard or commercially of the shelf parts and materials are purchased and they do not need any quality control. These components are generally entered into the institute with Quality Control Forms prepared by producers. These items are labeled as “No Need to Quality Control” and this type of components can be used in any test and production.

After most of the components are used in tests, they cannot be used in any other tests. However some of those components can have high value even after they are used and these components are required to be kept in stores. This type of components is labeled as “Valuable Scrap” and they cannot be used in tests and productions. However, these components can be utilized in areas such as training, illustration etc.

Some components are produced as trial for productions. The purpose of producing those is not to utilize them in products or tests. The quality of the production is tested with these components. If the components are not produced in required quality, components with higher quality are required from the suppliers. This type of components is labeled as “Trial”, and they cannot be used in tests and productions.

Produced components are put in stores in the institute. Each store has different places, called as cells, to put those components. The stores are specialized according to components types. While mechanical components are stored in one store, chemicals are stored in another one. There are also small amount of explosive materials and these are stored in highly secure stores. An inventory list is kept for each store. In this list, component, amount, and inventory entry date of those are listed. A sample inventory list is provided in Appendix C.

3.1.5 In-House Production Capabilities

Since the organization is engineer-to-order, diversified manufacturing and chemical processes are needed to produce items. However, in-house production capabilities are limited and only small portion of items can be produced in small volumes. In-house production capabilities are as follows:

Traditional Manufacturing Process

- Turning Operation (CNC)
- Milling Operation (CNC)
- Drilling Operation
- Band-saw Operation

Non-Traditional Manufacturing Process

- Electric Discharge Machining (EDM)
- Computer Aided Manufacturing (CAM with Unigraphics NX-4 software)

Chemical Production Capabilities

- Composite Solid Rocket Propellant Development and Prototype Production
 - Casting
- Plastic Bonded Explosive Development and Prototype Production
 - Casting
 - Slurry
 - Powder Compaction
- Flexible Linear Shaped Charge Development and Prototype Production
 - Slurry
 - Shape Forming
- Thermal Battery Development and Prototype Production
 - Dry Mixing
 - Fusion
 - Cold Pressing
- Pyrotechnic Igniter Development and Prototype Production
 - Powder Coating
 - Precipitation
 - Powder Compaction
- Inert Warhead Chemical Composition Development and Prototype Production
 - Powder Coating
 - Powder Compaction
- Incendiary Material
 - Powder Coating
 - Precipitation

3.1.6 Configuration Management

The development of a complex new system can be divided into a series of steps or phases. In each of these steps the system characteristics are defined as more specific system requirements and specifications. The integrity and continuity of these phases could be done by configuration management process (Alexander and Sweet, 2003).

A broad definition for configuration management is given by Hass (2003) as unique identification, controlled storage, change control, and status reporting of selected intermediate work products, product components, and products during the life of a system. A system component which physical and functional properties are going to be identified separately is treated as configuration item (CI) at TÜBİTAK-SAGE.

A narrow description of Configuration Management applied to Defense Systems is given by Kockler et al. (1990) as an integral part of system engineering management process for system definition and control in which the configuration management roles are defined as identification of functional and physical characteristic of selected system components, controlling changes to those characteristics, recording and reporting change processing and implementation status. The same guide states functions of configuration management as (1) Configuration Identification, (2) Configuration Control, (3) Configuration Status Accounting and (4) Configuration Audit. At any given time configuration management can provide description of CIs.

At SAGE, unique identification is realized by the Document Center by the help of project configuration manager. The process for unique identification starts with preparation of a new document. Documents can contain either engineering or production information and they are classified by usage areas. While some documents are prepared to describe parts used in the end products (developed products under projects) or product components, some other documents are prepared to describe auxiliary parts which are built as tools or equipments in order to produce the end products or product components. For all type of documents the process of the unique identification is the same. A new document is prepared and labeled by the unique number given by Document Center. The document goes to related project manager, subproject managers and work package leaders for approval and control. After the document is prepared and approved by managers and leaders, Document Center personnel notify all workers in the institute that the document is official and can be used and referenced at anywhere.

The unique identification number is formed from three parts. The first part named as prefix shows the document class and the length of this part is three digits. An example for prefix codes are “050” and “501”, the number “050” is used for engineering documents containing technical information about single mechanical unit. Numbers from “501” to “550” are used in production documents to describe chemical processes. The second part is six digits long and is called base number. The number is used for linking configuration related documents to CIs. All documents of a CI have the same base number. The third and final segment is three digits long and it is called as suffix code. This code represents the variation from the CIs. If there are two variations of the CI, “001” and “002” can be added as suffix code to the end of base number to represent each variations.

Controlled storage is realized by unique identification number with product batch (or lot) and serial numbers. At the institute, there are currently thirteen stores and each store is managed by a single manager. Each manager keeps the inventory list of his own store. Using this list, amount of the parts available in each store is calculated manually. Every item entering stores gets product lot and serial numbers. To consume or use a unit of item from store, everyone must fill an “inventory exit form” and the store manager drops the level of inventory by usage amount. By this way, the level of inventory records is kept up to date.

At SAGE, each item except procured parts and materials has three distinct set of numbers. Every item is produced or assembled according to a document. The document unique identification number given by the Document Center or project configuration personnel constitutes the first set of number. For every intermediate work products, product components, and products coming from the same production lot, a tree digit number is given to identify the lot. This number constitutes the second set of numbers. The last set of number is called as serial number and it composed of three digits and it is given incrementally to every unit in the lot. By the help of this set of numbers, controlled storage is accomplished.

Change control is a formal process to ensure that the change on a product, service or process is done in line with the needed modification. In this formal process, the impact of change is evaluated, and potential risks to overall system are considered. Work products are evolving through initial concepts to final products. In the initial phase of this period, change on work products is done by revision control which allows making changes rapidly. Revision control quarantines some safety backup copies and this provides a measure of control. At the end of the initial phase, the work product reaches to an acceptable state of completeness. At this time, the work product is accepted after a formal review and formal acceptance. Once a formal acceptance is done, no changes are permitted without formal change control. Formal change control with underlying revision control procedure provides a measure of stability. In this way, any changes to an accepted work product are carefully proposed, assessed, accepted and applied (Construx, 2006).

At SAGE, any change to a system component is requested by Engineering Change Proposal form. Any change proposal is evaluated by Configuration Control Board. This board is unique in every project and it is established to evaluate changes to system components in terms of operational effectiveness, cost, schedule, production and risk to system. The board can accept or reject the change proposal.

Status reporting of each CIs is done by configuration status list. These lists are kept for each CI and each list shows the changes on each CI. The list contains revision date, the person revising the document, the revision number and revision reason. Change tracking could be done on CI by using the CI status list.

3.2 Problem Definition

TÜBİTAK-SAGE develops diversified prototypes for different customers through R&D activities. Each customer has different requirements for each prototype. Since these prototypes are customer specific, completely different and unique prototypes are developed for each customer. For instance, while a system developed under a project is mostly based on electronic and software components, a system developed

under another project can be mostly based on mechanical and pyrotechnic components. The component variety in the institute is increased by developing customer specific prototypes. Currently, there are twenty-five ongoing projects in which each one has its own customer specifications. Each project has effects on component variety and the number of distinct components in the institute is increasing with each project undertaken. Besides, components are not standard and they are developed and modified in time. Therefore, there are numerous versions for each component. Table 3 displays the number of different components in four stores excluding versions of each component. The number of projects and their complexities increase day by day with a parallel increase in the number of components. The increasing number of components and continuous modification on these components causes problems related to material management. Each sub-problem is enumerated in the following paragraphs.

Table 3 Number of Distinct Components

Stores	2002	2003	2004	2005	2006
Store 1	253	293	378	479	576
Store 2	241	284	306	351	394
Store 3	213	259	382	468	597
Store 4	94	82	89	90	28

Various components have different characteristics and they can require different storage conditions such as specific temperature, humidity, electrostatic ground. This causes different stores to be built for specific group of components. There are currently thirteen stores in the institute and a manager is assigned for each one. Each manager is required to record entering and exiting dates of each component in the store. Inventory lists are kept in different files for each store. The files are prepared by word processors. To build a CI for a test, inventory lists of different stores have to be checked to find required parts. In addition to this, configuration lists have to be

checked to make sure the last variants of items is used. These files are stored as unlinked lists, causing inconsistencies.

Inventory lists are kept manually which mainly causes the following problems: These data are recorded by different managers, hence the data are not in good quality for calculation; while some quantities are stored as numeric formats without units, some quantities are stored as text with some notes. Therefore, someone has to work on these lists to calculate component quantities manually. Besides, the lists do not always have current inventory information. Therefore, the component quantities are not readily available and someone has to check amount of components in one store in a traditional way. Therefore, many problems in projects have been realized to find available and used components.

Production period of a product or component is ranged from days to years which usually managed by project phases. In each phase, components and developed systems are improved considerably. On one hand, components have different variants, on the other hand the end products have different variants. A variant can be differentiated from the original item in many ways and all life cycle of those variants are required in R&D activities. However, these data are kept in different files and formats. The information in these files cannot be used easily and takes considerable time of workers.

BOM is used for listing required components and their quantities required for a prototype system or subsystems. Since the configuration of a prototype is not static, each change in the prototype structure causes a different BOM to be created. Modifying the whole system's BOM can require a great effort, therefore a prototype's BOM is divided into many manageable subsystem BOMs. In this way, a small BOM for each subsystem is created separately and each change is carried out only on the related subsystem BOM. However, this process creates numerous and unlinked BOMs to exist in the system. This makes the calculation of material requirements for a prototype production a complex and problematic task.

Time estimation for prototype production is a challenging issue since it is necessary to calculate required number of components, time required to obtain lacking components and when these components are required. To produce a prototype, the lowest subsystems must be produced first, other subsystem are assembled in the order which is given in product tree. As doing planning, inventory must be checked for available items. The planning of this type of production needs high level of information flow between departments and requires a flexible tracking information system. At SAGE, the current information flow is problematic issue, since data is kept separately between divisions and data flow is manual and insufficient.

In R&D projects, BOM is changing continuously until the end of project. The components quantities are relatively small when it is compared to make-to-stock organization. However, contrary to make-to-stock organizations, the components are modified and the components varieties are relatively high. Current ERP packages are designed for make-to-stock organizations which make repetitive production of relatively small number of products. Therefore, these ERP packages cannot meet the requirements effectively in engineer-to-order organizations for material management purposes.

R&D activities require a great numbers of tests and prototype developments which require variable sized batch production of non-standard components. Since components are changing continuously, components cannot be produced or stored in large amounts. If a component fails or cannot meet requirements in one test, whole batches that component came can become useless or they have to be reworked to be used. To be cost effective, small sized batch production has been chosen previously. Since components are non-standard parts, lead time to acquire those components are generally long. Therefore, components have to be ordered a long time before they are needed. For this reason, acquisition of these components must be planned in a good way.

Configuration management is applied within projects. To realize configuration management, information is kept separately in different lists and formats. For

instance, KNAL (given configuration number list) is utilized for listing components of prototypes in product tree view. A sample KNAL list is given in Appendix D. In this view, no quantity information is kept. The required quantity information is kept in a different file that represent product tree in graphical tree format. These files must be linked to each other for calculation of material requirements.

Therefore, the data about configuration management and inventory management should be kept in a suitable and manageable way to be available for material management in prototype development. A system is required to access consistent information faster. An information system can be valuable and crucial for both integrating information consistently and accessing this information faster.

CHAPTER 4

PROPOSED SYSTEM

An information system is designed to store and integrate data about configuration management and inventories for material management purpose in the form of a product tree structure. The designed system has four modules. The modules and relations among these are given in Figure 11.

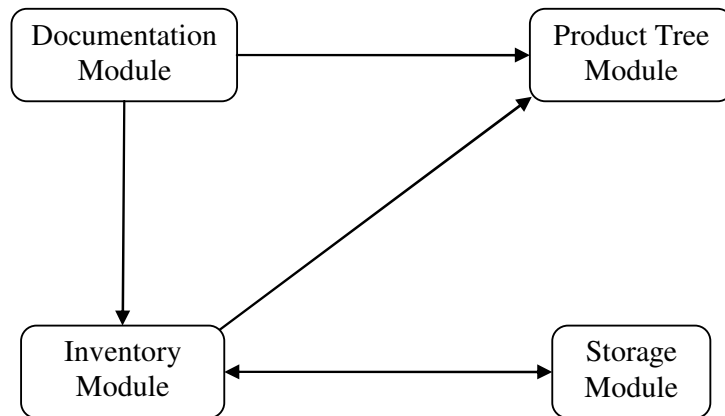


Figure 11 Integrated BOM System Modules

The documentation module is used for defining and managing information about documents. In order to define documents, the module does not need any information from other modules. The storage module is used for defining stores and cell places in each store. Items in each store or cell can be viewed and used via this module. The inventory module uses information provided by storage and documentation modules to enter items into the system. The module needs cell information provided by storage module to place items. This module is accessible through storage and product tree modules. The product tree module uses information provided by documentation

module to define products. The basic assumptions for the proposed system are given in the next section.

4.1 Basic Assumptions

In the proposed system, some basic assumptions are needed to make material management explicitly. The assumptions can be divided into two categories. The first group consists of assumptions related to time. The second group consists of assumptions related to material management in prototype production.

Time Related Assumptions:

- Time needed to perform each process is required to be known and can be represented as a linear function of quantity. This assumption also implies that each process has a standard duration which is required to be stable from one occasion to the other one.
- Times needed for assembling CIs are assumed to be known and can be represented as linear function of quantity. As for the process time assumption, the assembly time is required to be stable from one occasion to the other one.
- Lead times for procuring components are required to be known and can be represented as a linear function of quantity. Besides, any number of components can be ordered at the beginning of the planning period.
- All processes related to each CI are defined in the related documents. Time needed for performing undefined processes is assumed to be zero or negligible.
- Documents can have more than one version. Since processes are defined in documents, processes have also different versions. Time needed for performing these processes is assumed to be the same.

Material Management Assumptions:

- Unless all required CIs, parts and components required for a CI are not obtained, assembling of the CI cannot be started.

- The processes required to produce a CI are needed to be performed in series. After a defined process is completed, the next process must be started immediately.

4.2 Model-Base of the Proposed System

4.2.1 Documentation Module

The first step in the integrated BOM system is to define documents. Only information about documents those are especially prepared for CIs are needed for material management. The module stores and manages document related information not the document itself. Once information about a document is entered, the document can be used across different CIs and different projects. Since the module keeps information about documents, the module can be named as the document library.

Since each document can be revised in time, it is needed to keep information about the previous versions of the document. The first reason for this, there exist components which are produced according to previous versions of the document in the system. These items are accessed by the document and the version number in other modules. The second reason is that, a different version number for the new document must be given. The latter reason implies that a new version number must be compared with the previously given version numbers for a document. Therefore, the module keeps all the information about previous document versions, and makes it available to other modules.

Since the module controls and manages information about documents globally, some checks are required to be done before new documents or version of a document is added to the document library. The first check is done when adding a new document into the document library. The check controls whether the required information about document which are the document number, the version number and the name of the document are supplied. The second check is done when a new document is added to the document library and it checks whether any document is added previously with

the same document number. The final check is done when a new version for a document is added to the document library. This check assures that the same version of a document is not previously added to the document library.

4.2.2 Storage Module

The second step in the integrated BOM system is to define stores and cells in each store. Stores and cells are defined mainly by using their name. After stores are defined, cells in each store can be defined. Cells can be a shelf, box or any place to put items. Stores and cells are defined similarly and these two have similar properties such as width, length, height etc. but there are great differences between cells and stores. The first one is that stores can have many cells and cells can not have any other places. The second one is that, parts and components are put in cells in stores. Parts and components cannot be placed into stores without supplying cell information.

The module is designed for store managers. The responsible manager for a store can use this module to access available parts in the store. The items in a cell can also be accessible from the module. Not only the available items can be accessible, but also the used items can be accessed. The module uses inventory module to access information about available and used items in stores.

4.2.3 Inventory Module

The third step in the integrated BOM system is to enter general information about components. A CI exists in the system as a component on which a process is performed last. Therefore, each process related to CIs is defined in this module. Each process is explained in a related document. The document and version number are used in this module to enter component properties.

After components and processes are defined, each unique component can be entered into the system. Components are put in the defined stores and cells in storage

module. Components can be entered into the system one by one, but this would take considerable time. Therefore, the module also provides an option to enter items with the same properties as a whole. Whether the items are entered into the system one by one or as a whole, the module creates a record for each unique component. Since components in the same lot have similar properties, these components can be entered into the system by giving a range for serial number. For each number in the given series, a record would be created. If the items in the same lot have different properties like different quality result, these properties can be changed after components are entered into the system. As for the entering components into the system, components can be used one by one or as a whole. The designed system takes the repetitive work of entering and using components from the operators, and shortens the required time for these operations.

The module requires some information about the components to be entered by the operators. Therefore, the information provided is checked before being entered into the system. The first check is done when entering general information about components. The module checks whether the document related with the component is entered in documentation module previously. If the document is not defined in the documentation module, the operator cannot continue. The second check is done when components are defined. When components are entered into the system, the system checks whether the component was entered previously. So, the system prevents entering more than one record for the same component. The final check is performed when the components are used. The check controls quantity and if the component is not available in the system, the module does not allow the user to continue.

The lists for available and used components are kept separately in this module. When a component is entered into the system, it is added into available parts list and the amount of inventory for that item is set to given amount. When a component is used, the level of inventory for that component is decreased by the amount used. This amount is added into used part list and available part list is updated accordingly.

4.1.4 The Product Tree Module

The main function of this module is to maintain a product tree for each product. Bertrand and Muntslag (1993) proposed historical and project BOMs for engineer-to-order organization. These two BOMs are utilized in this module. Product tree has as a structure like multi-level BOM similar to VPC proposed by Olsen et al. (2001), but product tree is composed of CIs which require components and process performed items. Since each product is developed under a project, product trees are maintained under projects. In some projects, there are some variants of main product which are developed for mainly testing and measurement purposes. Product trees for these variants are also maintained along with main product trees in this module. This module can create product trees for variants (variant product tree) using product trees which are maintained for main products (main product tree). The module can use main product tree in time to create variant product trees. After a variant product tree is created, it can be modified without affecting the other product trees.

The operations on product tree can be divided into two groups. The first group contains editing operations which are used for adding, deleting and moving CIs in multi-level BOM structure. The second group of operations contains linking operations which are used for linking elements in other modules to CIs. The required processes and components, which are defined in documentation and inventory modules, are linked to the related CIs. After processes are linked to related CIs, order of processes to produce CIs can be specified. The other linking operation is for predecessor type relation. In order to produce a CI, a group of CIs is required to be completed previously. This group of CIs can be linked to the former CI as a predecessor.

This module uses information entered from storage, documentation and inventory modules to calculate material requirements and time required for production. In order to calculate material requirements, required CIs and their required quantities are calculated first. To do this, product tree is traversed starting from level zero (level of the main or the end product) to the last level. The main product quantity is given by

the user. Quantities of CIs in level one are calculated next. CIs in level two are calculated next by using quantities in level one. This procedure is applied until required quantities of CIs in the last level are calculated. After CI quantities are calculated, the required quantities of components and process performed items are calculated. The module uses a network based model to find time needed for assemblies and subassemblies. The network model is explained in chapter five by illustrative example. Since the module can use information entered from inventory module, available quantities for components in stores are displayed with required quantities. The lack and excess amounts are calculated using quantities of available components and required.

A main product tree contains many CIs and their product trees implicitly. The module can extract a product tree for each one of these CIs. After product tree for a CI is extracted, the module can calculate material requirements and needed time for the CI using the extracted product tree. The same procedure for calculation is applied to the extracted product tree.

Since the end product and the other items in the product tree are all CIs, any change in the product tree is required to be recorded. Therefore, each editing and linking operation is recorded with event time. Using these records, the module can create and reach the product tree and its links with other modules at any point in time. The new product tree is created as a variant of main product tree. So, the module keeps only one main product tree, and can reach to all product trees (historical BOMs) in time without storing much data. In the traditional methods, all versions of product tree are stored with the required documents' information. Therefore, a great amount of data is stored in the traditional method. The designed system decreases the amount of data stored for product trees by creating or reaching them dynamically.

The module requires from user to enter CIs in the product tree, the required processes and components to produce each CI. Therefore, some checks are required to be done before entering information into the system. The module does not allow users to link a process or a component that is not defined in inventory module. While precedence

relations are being defined, the module checks for cycling. The cycling case is given in Figure 12. In the figure, each one of CI 1 and 2 are required for producing CI 3 and these CIs cannot require CI 3 to be produced previously. The predecessor links displayed as dotted lines in Figure 12 creates cycling.

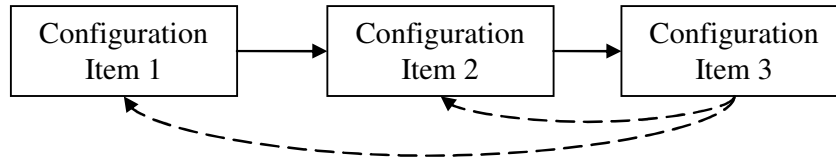


Figure 12 Cycling Relations

4.3 Database

The proposed system needs to reach and manage great amount of data about documents, components and product trees. Therefore, a relational database using Microsoft Access is designed as a basis for the proposed system. After performing some checks on user provided data, the system enters data to the tables in the database. The system uses data in the relational database, and calculates material requirements and then makes time planning for any CI production.

To design database, entities in the system are determined first. The main entities in the system are:

- Stores
- Store Cells
- Items
- Documents
- Document Versions
- Projects
- Product Trees

- Product Tree's Versions
- Product Tree's Variants
- Variant's Versions

The one-to-one and one-to-many relationships between entities are determined next. According to determined relationships, the entity-relationship diagram is presented in Figure 13. The relationship diagram provided by Microsoft Access for designed database is given in the Appendix E.

4.3.1 Database Tables

The tables in the designed relational database are:

Projects: All Projects completed or continuing are stored in this table. This table is composed of Project Number, Project Name, Project Start and Project Finish fields. The primary key for the table is Project Number field.

Document Scopes: Document scopes are defined in this table. This table is composed of Scope ID and Scope Name fields. The primary key for the table is Scope ID field. Since Document Scopes are stored in table, any new scope can be entered into the system without any change in the proposed system coding.

Document Types: Document types are defined in this table. This table is composed of Type ID and Type Name fields. The primary key for the table is Type ID field. Like Document Scopes, any new document type can be entered into the system without changing proposed system coding.

Documents: Information about documents that are especially prepared for production of CIs is stored in this table. This table only holds the information about the last version of documents. The table is composed of Document ID, Scope ID, Type ID, Document Number, Version Number, Document Name, Preparation Reason, Person Prepared, Preparation Date, Preparation Note fields. The primary keys for the table are Document Number and Version Number fields.

Document Changes: When a document is required to be changed, a new version of the document is prepared. Hence, this table holds information about each version of the documents which are defined in Documents table. The table is composed of all the fields of Documents table and additional two fields which are Event Type and Event Date. The primary keys for the table are Event Type and Event Date fields. There are three values for Event Type field which are Add, Update and Delete. When a document is added to Document table, the proposed system also records the document as Add Event in Document Changes table with time of the event. If the document in Documents table is updated, this event with the document information is recorded as Update Event. The last event occurs when a document in Documents table is deleted. This table is required since when a document in Document table is updated or deleted, the information about a document before event is kept in this table and the information about updated and deleted documents is not lost and the system can reach information about these documents by using this table.

Quality Results Types: Some items are required to be controlled in terms of their quality. This table provides type definition to these items. These types are not included in proposed system coding and are defined and taken from this table. Hence when a new type is required to be defined, the type can be added to this table without changing proposed system coding. The table is composed of Code, Definition and Type Note fields for quality result types. The primary key for the table is Code field.

Item Properties: General properties for items are stored in this table. The distinction between components and process performed items is made in this table. Some items are needed especially for a specific project but can be used in any project. Therefore, the default project value for items is stored in this table. The table is composed of Type, Project Number, Item Number (or Document Number), Item Name, Unit, Compatibility Class, MSDS number, Hazard Class, Item Note, Fixed Day Required (for each order to obtain item), Slope Day Required (additional time to obtain each unit of item), Enter Date (when Item Definition entered into the system). The primary key for the table is Item Number field.

Stores: Storage information is kept in this table. The table is composed of Store ID, Store Name, Store Manager, Open Date, Close Date, Maximum Usage Area, Store Note, Height, Width and Length fields. The primary key for the table is Store ID field.

Store Cells: The places in stores are kept in this table as Store Cells. The table is composed of Store ID, Cell ID, Cell Name, Cell Capacity, Cell Type, Cell Note, Open Date, Close Date, Height, Width and Length fields. The primary key for the table is Cell ID field.

Items Available: Available items in the stores are recorded in this table. The proposed system requires Item Inventory Property, Store and Cell information to be entered into the related tables previously. When the items are used, the quantity values for items are decreased. If quantity of an item is decreased to zero, the item is deleted from Items Available table. The table is composed of Project Number, Item Number, Version Number, Lot Number, Serial Number, Quality Result Code, Cell ID, Quantity, Production Date, Due Date, Work Order Number, Budget Number, Inventory Enter Date, Inventory Note fields. The primary keys for the table are Item Number, Version Number, Lot Number and Serial Number fields. Unit of the items are stored in Item Inventory Properties table. The user is required to enter item quantities in terms of entered units.

Items Used: Information about used items is stored in this table. Only the items in Items Available table can be used. The table is composed of Project Number, Item Number, Version Number, Lot Number, Serial Number, Quality Result Code, Cell ID, Quantity, Production Date, Due Date, Inventory Work Order Number, Use Work Order Number, Budget Number, Inventory Enter Date, Inventory Exit Date, Items Used By, Inventory Note, Inventory Use Note fields. The primary keys for the table are Item Number, Version Number, Lot Number, Serial Number and Inventory Exit Date fields. Like Items Available table, unit of the items are stored in Item Inventory Properties table. The user is required to enter used item quantities in terms of entered units.

Product Trees: The table contains product trees and variant of these product trees prepared for each project. There is at most one main product tree for each project. The table can contain any number of variant product trees for projects. The proposed system can create variant product trees from project main product tree. The table is composed of Project Number, Variation ID, ID, Above ID, Below ID, Left ID, Right ID, Product Tree Number, Item Name, Base Number, Quantity, Fixed Day Required (for assembly operation), Slope Day Required (for each assembly operation), Start Date, Item Note fields. The primary keys for the table are Project Number, Variation ID and ID fields. Product Tree Number is used for locating a CI in the product tree. To manage product trees and form Product Tree number the ID values explained below are utilized:

- *ID:* A CI in each product tree has an ID number. This number is unique and the CI can be identified by this number.
- *Above ID:* Any number of CIs (child items) can be required to produce another CI (parent item). The ID of Parent Item is stored as Above ID in the child Items' records.
- *Below ID:* When the child items are sorted and indexed starting from one, ID of the child item with index value one is stored in the parent item's record as Below ID.
- *Right ID:* There can be more than one child item in any level. The location of child item in the level can be specified by the help of Right ID value. When the child items are sorted on a line from left to right, there would be one child item at right of each child item, unless the child item is located at the right end of the line. The right ID value of each child item record contains other child item's ID which is located closest right of each child item.
- *Left ID:* The left ID value is utilized with Right ID value to locate child items in any level. When the child items are sorted on a line from left to right, there would be one child item at left of each child item, unless the child item is located at the left end of the line. The left ID value of each child item record contains other child item's ID which is located closest left of each child item.

When the child items for each parent item are sorted on a line using left ID values and indexed starting from left to right, the product tree number can be formed by adding index value to the end of parent item's product tree number and separating index value and product tree number by using one dot.

Product Tree Changes: The proposed system stores any editing operations on Product Trees in Product Tree Changes table. The table is composed of all the fields of Product Trees table and additional two fields which are Event Type and Event Date. The primary keys for the table are Event Type and Event Date fields. There are five values for Event Type field which are Add, Update, Delete, Move Add, Move Delete and Version Add. Add, Update and Delete values are self explanatory. When a user moves a CI in product tree, two records are stored in Product Tree Changes table for this moving operation. The location where the CI moved from is recorded as Move Delete event, and the location where the CI is moving to, is recorded as Move Add event. The Version Add event stores the current time to Product Tree Changes table for one product tree when a user adds a version. After this event is performed, version of product tree is increased by one. The proposed system uses events and event times in this table to create a variant of a product tree.

Product Tree Variations: General information of each created variation for a product tree is stored in this table. The table is composed of Project Number, Variation ID, Variation Name, Variation Date (create time of a variation) and Variation Reason fields. The primary keys for the table are Project Number and Variation ID fields. The table stores also general information for each main product tree. The zero value for Variation ID field is used for main product trees. So, when a user wants to create main product tree for a project, the proposed system checks first whether there is a zero value in variation ID field for that project in this table.

Configuration Item Requirements: The proposed system stores documents, components and processes which are linked to CIs in Configuration Item Requirements table. The related documents, components and processes are defined in other tables. This table only holds linking information supplied by linking operation

on Product Trees. The table is composed of Project Number, Variation ID, Configuration Item ID, Required Item Number (Item or Document Number), Quantity, Order Number fields. The primary keys for the table are Project Number, Variation ID and Configuration Item ID fields. The order of processes on a CI is stored in this table by using order number field.

Configuration Item Requirements Changes: All changes in Configuration Item Requirements table are stored in this table. The table is composed of all the fields of Configuration Item Requirements table and additional two fields which are Event Type and Event Date. The primary keys for the table are Event Type and Event Date fields. Event Type can be one of Link, Unlink, Update, Move Up and Move Down values. Link, Unlink and Update values are self explanatory. The order of process given in Configuration Item Requirements table can be changed. When a process is taken to a prior place in the order, this event is recorded as Move Up event. When a process is taken to a succeeding place in the order, this event is recorded as Move Down event.

Configuration Item Predecessors: The CIs which have to be completed previously to produce other CIs are stored in this table. The table is composed of Project Number, Variation ID, ID (of CI), Predecessor ID (ID of CI which have to be completed previously) fields. The primary keys for the table are Project Number, Variation ID, ID and Predecessor ID fields. The values in this table are used when planning network is being constructed.

4.3.2 Normalization

Normalization is the restructuring process of the logical data model of a database. With this process, redundancy can be eliminated, data can be organized efficiently, repeating data can be reduced and the potential anomalies during data operations can also be reduced. Data normalization can improve data consistency and simplify future extension of the logical data model (Wikipedia, 2006). There are different levels of normalization which are named as normal forms. The requirements of the first three levels are as follows:

The database is in the first normal form if following are realized:

- Each table has primary key.
- Repeating groups are eliminated.

The database is in the second normal form if following are realized:

- Requirement for the first normal form are met.
- In this normal form, each column must depend on the entire primary key.

The database is in the third normal form if following are realized:

- Requirement for the second normal form are met.
- Each column must depend on directly on the primary key.

The developed database in the third normal form since, no repeating column exists and all columns depend directly on the primary key in the tables.

4.4 User Interface Design

In the previous sections, the modules and database tables in the developed system are explained. The general flowchart for the designed system is given in Figure 14. The user interfaces provides an environment for users to manage data in tables when performing defined operations.

Store Definition is entered from store module. The operation enters data into two tables. The defined store is entered into Stores table. The operation creates also a blank cell for the defined Store. The blank cell information is stored in Store Cells table. If a store has any cell, item or used item, then the store cannot be deleted from the designed system.

Cell Definition is entered from store module. In order to enter cell information, first a store for the cell must be provided. The defined cells are stored in Store Cells table. If a cell has any item or used item, the cell cannot be deleted.

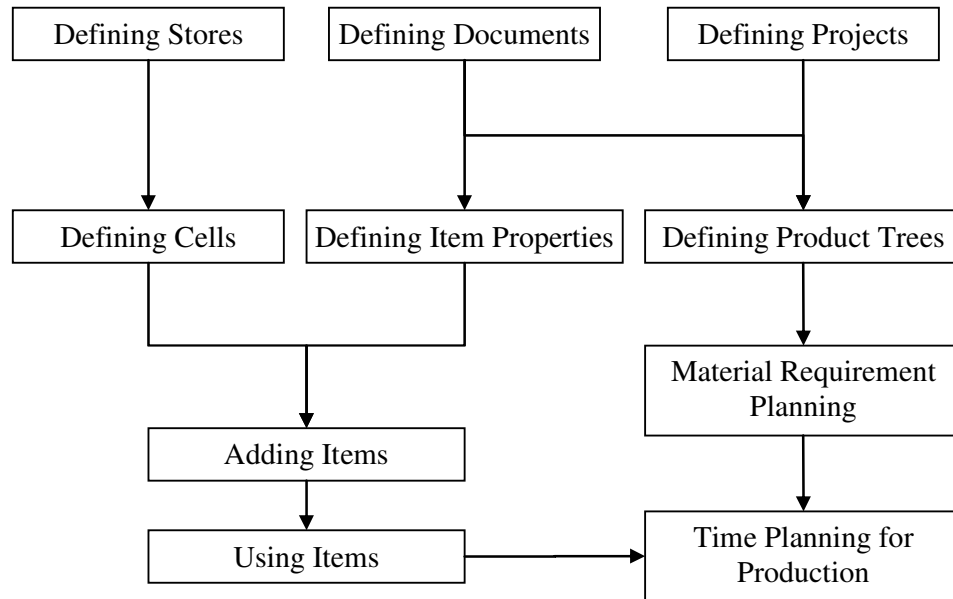


Figure 14 General Flowchart for the Designed System

Documents are defined in documentation module. The operation uses Documents and Document Changes tables. When a new document is being defined, the document information is added into Documents table. The Document is also recorded as Add Event to the Document Changes table. After a document is defined, document versions can be entered into the system. When adding document versions, the document information in Documents table is updated, and the version information is added into document changes table as Update Event. A document can be deleted from document library, but information about the deleted document can be accessible from Document Changes table. When a document is deleted from document library, the document cannot be revised any more.

After documents and projects are defined, the user can define item properties including process information from the inventory module. The defined items are entered into Item Properties table.

After item properties and cells are defined, the items can be added into the designed system. Inventory module is designed for Adding Items to the system. Added items are stored in Items Available table.

After Items are added into the designed system, the added items can be used from inventory and product tree modules. Using Items operation utilizes Items Available and Items Used tables. When items are used, quantities of items are decreased from Items Available table. The used amount is added into Items Used table.

After documents and projects are defined, product tree module can be used for defining product trees. This operation uses Product Trees, Product Tree Changes, Configuration Item Requirements, Configuration Item Requirement Changes and Predecessors tables. Main product trees or variants of these product trees are all stored in Product Trees table. Changes in main or variant product trees are stored in Product Tree Changes table. Changes in product trees are stored as event in Product Tree Changes table. Items required for CIs in product trees are stored in Configuration Item Requirements table. When a change occurs in the configuration item requirements, the change is recorded in Configuration Item Requirements Changes table as an event.

Material requirements for an assembly or subassembly are calculated in the product tree module. The operation does not affect any table. The product tree structure is used along with configuration items requirement to find the necessary amounts of components and CIs. This operation calculates also time to obtain lacking items.

Time planning for production is done from product tree modules. This operation does not affect any table. The operation uses the planning model given in Chapter five for time planning. Information in all tables is used for time planning for production.

The integrated BOM system is needed to work in intranet environment to enable users to enter the system simultaneously. Therefore, the user interfaces are prepared in ASP.NET environment by using C# as coding language. The user interface

screens can be reached from any web browser. The user screens are given in Appendix F.

4.5 Benefits of the Proposed System (Verification)

There are thirteen inventory lists in the institute currently. In the proposed system, the data in the inventory lists are kept in two tables with the same format. The inventory files are eliminated in the proposed system. The number of files in the proposed system is decreased by the number of inventory files.

A great number of files for BOM are utilized in each project currently. The BOM for the whole system is divided into small manageable BOMs for subsystems. In one project, the BOM for the whole system is divided into seven main subsystems in which each one has been revised ranging from two to eight times. Therefore, the number of BOM reaches to thirty-two BOM files in this project. Each version is kept in a separate file. These files are eliminated by dynamic BOM structure and only one BOM is maintained under a project in the proposed system. These BOM structures are kept in Product Trees and Product Tree Changes tables.

A KNAL is kept for each of the six main subsystems. These files display status of each CI in the subsystems. Six different files are modified ranging from 19 to 269 times. Each version of these files is stored. Therefore, total number of KNAL files reaches to 809 files. These files are eliminated in the proposed system by utilizing Configuration Item Requirements and Configuration Item Requirements Changes tables.

The number of files is decreased definitely and data in different files are integrated in the proposed system. This increases data consistency and reduces data redundancy. Different formats used previously for storing data are eliminated in the proposed system.

CHAPTER 5

THE PLANNING MODEL

In this chapter, the model used for time planning for production is explained by an illustrative example. The first section defines a valid product structure by using a simple product with simple components and processes. The next section then explains the construction of network for planning purposes and gives the CPM/PERT solution to the presented network.

5.1 The Product Structure

To produce a product, various components and CIs are assembled. The simple product, which is also a CI, is produced using five CIs. The product structure with required quantities for each CI is presented in Table 4. Usually, products start with a small number of CIs and reach to more than a hundred CIs at the end. Since a product is a CI, a change in product structure is always controlled and recorded. One of these recorded structures or the current structure can be used as a starting product structure for creating product variants to be used in tests or developing deliverable product variants to customers.

Table 4 Simple Product Structure

Product Tree Number	Name	Required Quantity
01.	CI 1	1
01.01.	CI 2	4
01.02.	CI 3	3
01.02.01.	CI 4	3
01.02.02.	CI 5	4
01.03.	CI 6	2

In Table 4, product tree number displays which CI is assembled from other CIs in which level. The number can be split into two parts. All the digits but the last two ones form the first part which shows where the CI is used. The second part, which is formed from the last two digits, displays CI order in product tree level. According to the given product tree, CI 1, which is the end product, is assembled by using four units from CI 2, three units from CI 3 and two units from CI 6. The CIs used for assembling CI 3 can also be extracted from the given product structure. To produce one unit from CI 3, three units of CI 4 and four units of CI 5 are used. The product tree structure given in Table 4 is presented in Figure 15 with required quantities given in parentheses.

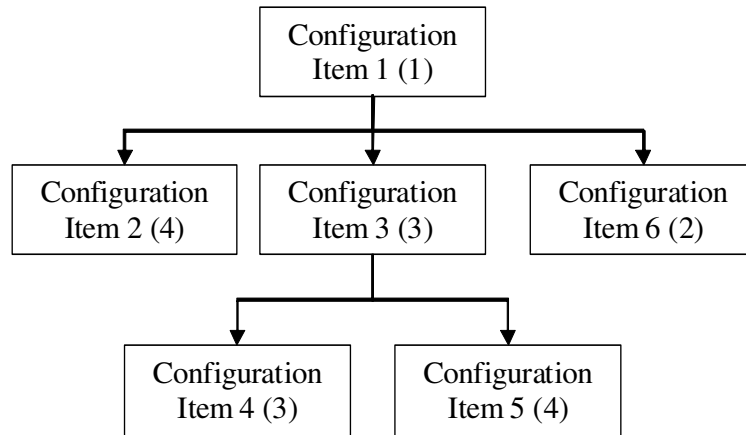


Figure 15 BOM for the Simple Product

A CI can require performing different processes and acquisition of components. After all processes are performed on the initial version of CI, any number of components and CIs are assembled to form the final version of CI. When there are no required components or other CIs to form the final CI, an assembly operation is not needed. General production process of a final CI is given in Figure 16. T1 is the time required for all of the defined processes to be completed on the initial CI. These processes for the CI are performed sequentially. Therefore, T1 is calculated by

summing all of these processes' times. T2 is the time required for acquisition of components and production of CIs. T2 is the maximum among lead time of Component A, lead time of Component B or production time of CI A when none of these items are available in store. Operations performed during T1 and T2 are parallel to each other. T3 is the time required for assembling the final CI if it is required. Assembly operation can not start until all of the required components or CIs are obtained and all processes on the initial CI are performed. Therefore T3 is serial to T1 and T2. Total time required to produce a final CI can be found by adding T3 to the maximum of T1 and T2.

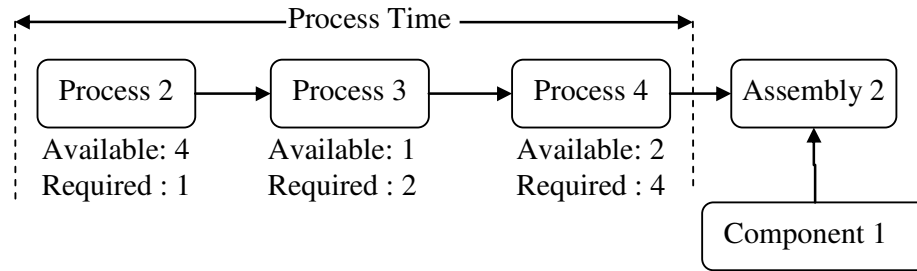


Figure 16 Production of the Final Configuration Item

For simplicity, it is assumed that there are three components available in the system to produce CI 1. Components are standard parts and materials that can be procured from suppliers. These components can be kept in store to be used in production of CIs. The component list with inventory levels and lead time formulas are given in Table 5. Components and CIs available in store are used for production first, and when there is a shortage of these items, components are procured and CIs are produced to be used in assembly. To explain the model, integer values are used in the given formulas and quantities but rational numbers can be used in them as well.

Each process is described by a configuration document and the document is specific to a CI. Therefore each process is unique and specific to a CI. When a process is performed on an item, the item number is changed to indicate the performed process.

To produce CI 1, the required processes by each CI and time required to perform these processes are given in Table 6. After a process is performed on an item, the item can be placed in store to be used in production later. A process is performed on an item only once. Since productions of initial CIs are performed according to a configuration document, this can be modeled as a process. Therefore, at least one process for CIs must be defined. Since each process is unique and is defined by CI number, each process given in Table 6 is different and is used in a single CI.

Table 5 Component List with Lead Times

Component Name	Available in Store	Lead Time Formula
Component 1	30 units	$4+3Q$ (day)
Component 2	40 units	$6+3Q$ (day)
Component 3	12 units	$8+5Q$ (day)

Table 6 Process List with Process Times

Configuration Item	Process Name	Process Time Formula
CI 1	Process 1	$1+6Q$ (day)
CI 2	Process 2	$1+7Q$ (day)
	Process 3	$8+1Q$ (day)
	Process 4	$4+2Q$ (day)
CI 3	Process 5	$2+3Q$ (day)
CI 4	Process 6	$5+2Q$ (day)
	Process 7	$4+3Q$ (day)
CI 5	Process 8	$6+3Q$ (day)
CI 6	Process 9	$0+3Q$ (day)

The time values in Table 5 and Table 6 are given as a linear function of unit quantity. The time is assumed to be given as day but another time unit like hours and weeks

can also be used, but each formula must be given in terms of the same time unit. Available WIP (work in process) CIs are given Table 7. Components and their required quantities for each configuration item are given in Table 8.

Table 7 Available WIP CIs

WIP CIs	Available in Store
CIs with the last process 2	4
CIs with the last process 3	1
CIs with the last process 4	2
CIs with the last process 5	1
CIs with the last process 6	5
CIs with the last process 9	10

Table 8 Component Requirement for CIs

Configuration Item	Required Components	Required Quantity
CI 1	Component 3	2
CI 2	Component 1	1
CI 5	Component 1	3
	Component 2	4

According to production model of CI given in Figure 16, an assembly operation for a CI is needed when the CI requires at least one component or at least one other CI. CIs 1 and 3 require other CIs, whereas CIs 2 and 5 require at least one component. Therefore an assembly operation is required for CIs 1, 2, 3 and 5. To produce CI 1, the required assembly operations for CIs are given in Table 9 with their respective time. Since the rest do not require assembly operation, assembly time for them is zero.

Table 9 Assembly Operations

Assembly Name	Time Formula
Assembly 1	$2+4Q$ (day)
Assembly 2	$3+2Q$ (day)
Assembly 3	$3+1Q$ (day)
Assembly 5	$4+1Q$ (day)

Some CIs need another CI in any level in product tree to be completed in advance. To produce the end product, CI 2 needs to be completed before the assembly of CI 3 and likewise CI 4 has to be completed before the assembly of CI 6.

The production process of CI 1 is presented in Figure 17. The component requirements are aggregated in this production process. Aggregation aims to encourage procuring components at the beginning of the production by a single order. In this way, lead time of component acquisition would be less than it would be when components are ordered separately. After required components are obtained at the beginning of production, only process and assembly operations are left behind to produce CIs.

The required amount of components and CIs can be found easily by traversing BOM. However, the calculation of the time required for production of CI 1 with current inventory level is not easy. In this case, the solution to find the required time for production is to utilize CPM/PERT method. In order to use this method, a network must be created. The following section describes how the network is constructed.

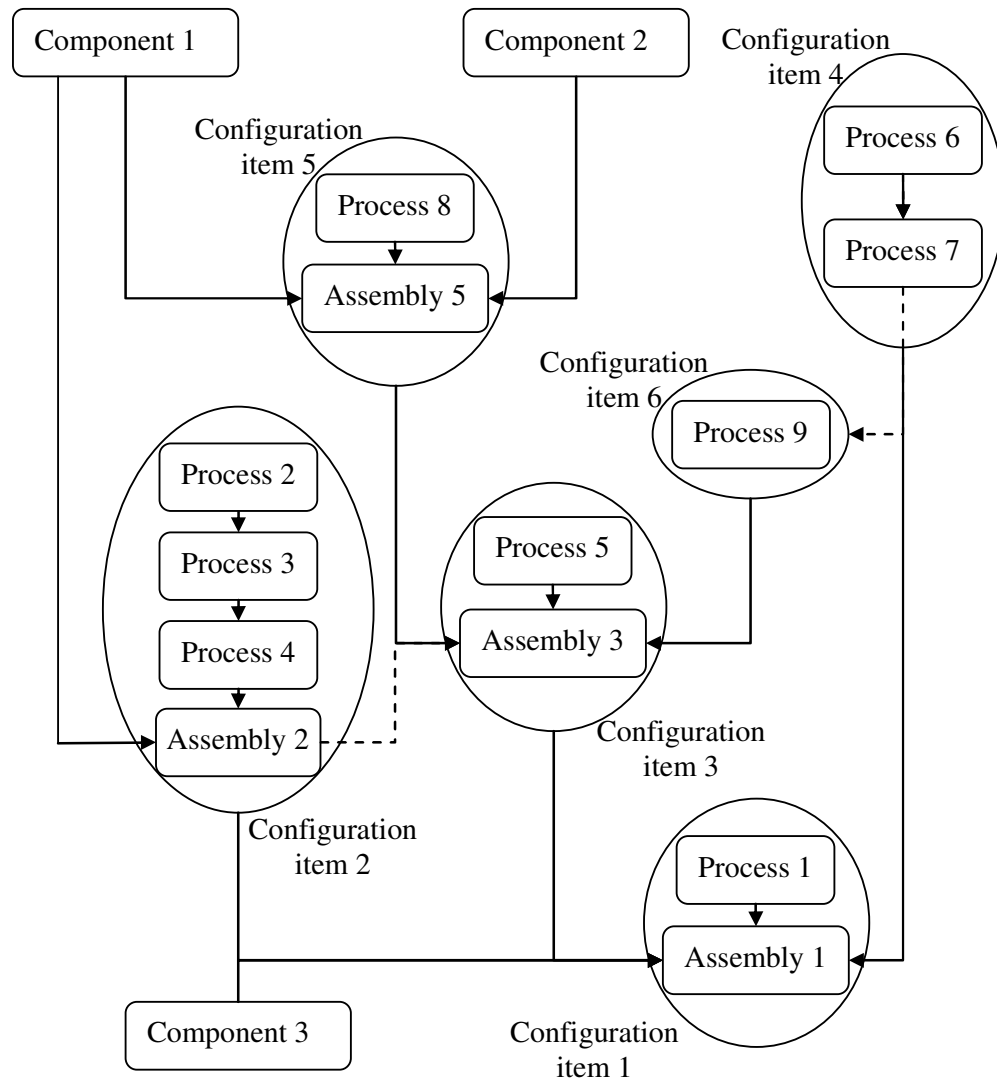


Figure 17 Production Process of the Configuration Item 1

5.2 The Network

To calculate the production time of CI 1, an activity-on-node network is used. First, dummy start node is included in the network with zero duration, and then nodes are created for acquiring components, performing processes and assembling items. The precedence relations between created nodes are given to form the network. The

CPM/PERT solution to this network is presented with early and late start and finish times. Lastly slack times are calculated for each node.

The items given in Table 7 can be divided into two groups. The first group of items is finished CIs on which the last operation defined for CI is performed. The second group contains the items which require additional operations to be performed for obtaining finished CIs. Total quantities for CIs are given in Table 10. The required quantities for CIs are calculated by traversing BOM and are also given in Table 10.

Table 10 CIs Quantities

Name	WIP quantity	Finished quantity	Total Quantity	Required Quantity
CI 1	0	0	0	1
CI 2	5	2	7	4
CI 3	0	1	1	3
CI 4	5	0	5	9
CI 5	0	0	0	12
CI 6	0	10	10	2

Total requirements for components can be calculated using Table 8 and Table 10. The needed quantity for each component is calculated by summing each requirement coming from each CI. Table 10 gives required quantity for each CI and Table 8 gives component requirements for each CI. The required component quantities are calculated by first multiplying required quantities of components for the respective CI given in Table 8 with required quantities of each CI given in Table 10 and then summing up the resulting component quantities. The required component quantities are given in Table 11 along with available quantities and shortages for each component type.

Table 11 Component Quantities

Component Name	Available in Store	Required Quantity	Shortages
Component 1	30 units	40 units	10 units
Component 2	40 units	48 units	8 units
Component 3	12 units	2 units	0 units

For each component type which is lacking in store, a node is necessary to represent component acquisition. Since, these nodes are primarily related with components, they can be named as component nodes. Duration of component nodes can be found by using Table 5 and Table 11. Component nodes and their duration are tabulated in Table 12. The predecessor of component nodes is the dummy start node, which has node 0.

Table 12 Component Nodes Duration

Node Number	Component Node for	Duration	Predecessor
1	Component 1	34 days	0
2	Component 2	30 days	0
-	Component 3	0 day	-

For each CI, a node is necessary to represent the processes performed on the CI. Since these nodes are primarily related with processes, they can be named as process nodes. When there are no finished or work-in-process CIs in store, duration of these nodes is found by summing all defined process times for a CI. When there are sufficient amount of finished CIs to cover required quantity, the duration of process nodes become zero. Getting all finished CIs from store for assembly operation is assumed to be negligible. When there are no sufficient finished CIs in store, work-in-process CIs are completed by performing required processes on them. Work-in-process items requiring fewer processes are used first. The duration of process nodes

are calculated by summing required process times. The predecessor of these nodes is dummy start node. As an example, Duration of process node for CI 2 is calculated by utilizing Figure 18.

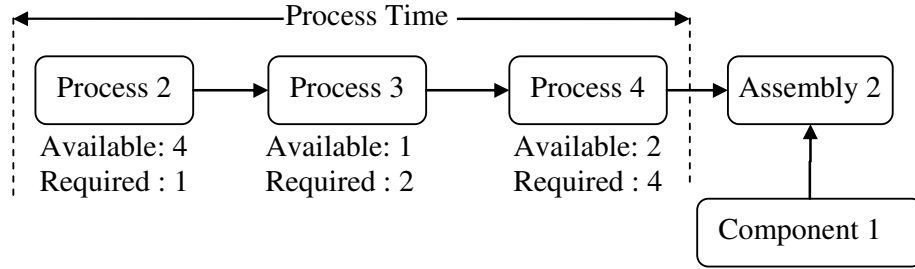


Figure 18 Duration of Process Node for CI 2

Since, producing CI 1 requires four units from finished CI 2, four Process 4 performed items (i.e. items on which Process 4 is performed last), are required. However, only two Process 4 performed items, are available in store. Therefore, the remaining items are tried to be covered from Process 3 performed items. While two Process 3 performed items are needed, only one is available. One required quantity is covered from Process 3 performed items. The missing items are tried to be covered from Process 2 performed items. Only one item is required and four units are available in store. One unit from Process 2 performed items is used for covering the missing quantity. When the aforementioned procedure is considered, it is seen that time is needed to perform required processes for changing Process 2 and Process 3 performed items to Process 4 performed items. One unit from Process 2 performed item requires Process 3 and Process 4 to be performed. In the same way, one unit from Process 3 performed item requires Process 4 to be performed. So, one Process 3 and two Process 4 must be performed on those items. Time can be found by using Table 6. Nine and eight days are required for performing Process 3 and Process 4 respectively. The duration of process node for CI 2 is seventeen days. Calculated duration of process nodes for CIs are calculated in the same way and tabulated in Table 13.

Table 13 Durations of Process Nodes for CIs

Node Number	Process Node for	Duration	Predecessor
3	CI 1	7 days	0
4	CI 2	17 days	0
5	CI 3	8 days	0
6	CI 4	44 days	0
7	CI 5	42 days	0
8	CI 6	0 days	0

For each CI, whether it requires assembly operation or not, a node is created to represent assembly operation. Since, these nodes are primarily related with assembly operation, they are named as assembly nodes here. If CI does not require assembly operation, duration of the assembly node is taken as zero. The predecessor of assembly nodes are the related process nodes, the related component nodes and assembly nodes of required CIs. Duration of assembly nodes are calculated by using Table 9 and Table 10. Calculated durations for assembly nodes are listed in Table 14.

Table 14 Durations of Assembly Nodes for CIs

Node Number	Assembly Node for	Duration	Predecessor(s)
9	CI 1	6 days	3, 10, 11, 14
10	CI 2	11 days	1, 4
11	CI 3	6 days	5, 12, 13
12	CI 4	0 days	6
13	CI 5	16 days	1, 2, 7
14	CI 6	0 days	8

In some cases, some CIs require other CIs to be completed in advance. This situation can be handled in the network diagram by setting a predecessor between related

nodes. For the production process given in Figure 17, two predecessors are formed between CIs. The first one is between CI 2 and CI 3. Assembly node of CI 2 becomes predecessor for assembly node of CI 3. Similarly, the second one is set between CI 4 and CI 6. Assembly node for CI 4 becomes the predecessor of assembly node for CI 6. The added predecessors for assembly nodes are given in Table 15 in bold.

Table 15 Configuration Item Precedence Relation

Node Number	Assembly Node for	Duration	Predecessor(s)
11	CI 3	6 days	5, 10 , 12, 13
14	CI 6	0 days	8, 12

The required time for production of CI 1 is the time between start time of dummy start node and finish time of assembly node of CI 1. The network diagram for the nodes given previously is presented in Figure 19.

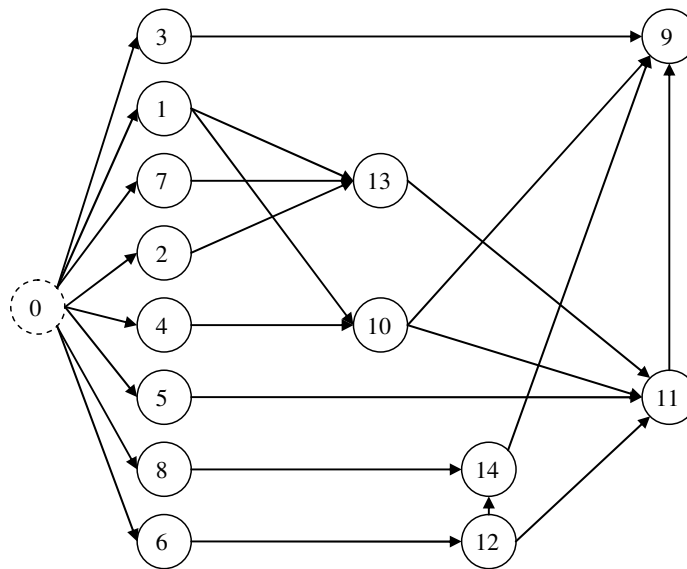


Figure 19 Network Diagram for Production Process

The CPM/PERT solution to the network given in Figure 19 is presented in Table 16.

Table 16 CPM/PERT Solution to CI 1

Node Number	Duration	Early Start	Early Finish	Late Start	Late Finish	Slack
1	34	0	34	8	42	8
2	30	0	30	12	42	12
3	7	0	7	57	64	57
4	17	0	17	30	47	30
5	8	0	8	50	58	50
6	44	0	44	14	58	14
7	42	0	42	0	42	0
8	0	0	0	64	64	64
9	6	64	70	64	70	0
10	11	34	45	47	58	13
11	6	58	64	58	64	0
12	0	44	44	58	58	14
13	16	42	58	42	58	0
14	0	44	44	64	64	20

Table 16 can be used for time planning for production. The table shows start and finish time of each assembly. According to the table, the prototype can be assembled in seventy days (early finish of assembly node 9) with the current inventory levels. The table also gives the slack values for each assembly. CI 2, CI 4 and CI 6 assembly operations have slack values. Therefore, these assemblies can be delayed without changing the assembly duration of CI 1.

In this model, durations are assumed to be known with certainty. Uncertainties about durations can also be incorporated into the model. The best, the moderate and the worst case durations can be utilized for uncertain durations. In that case, the planning model can provide assembly time for the best, the moderate and the worst cases.

CHAPTER 6

CONCLUSION

In this thesis, an information system is developed for material management in an engineer-to-order organization which develops unique, high valued and complex prototypes. The developed information system handles variations of numerous prototypes and components in an efficient way. A database is developed for effective integration of inventory and configuration management information, and a model is included to the developed system to prepare a time plan for production.

In the system development process, prototypes, composed of various components, are required to be manufactured. Due to the nature of the process, technical requirements of these prototypes and components are changing or evolving continuously which brings the need for an effective material management system. In this process, after components and prototypes are designed, these components are made to be manufactured by selected suppliers. These components are subjected to quality control by the institute prior to their acceptance and usage. The components that satisfy quality requirement are assembled and tested in the institute. The results of these tests give feedbacks to design phase and components are changed accordingly. A huge amount of information is required to be documented and stored by different divisions during this process.

In this thesis, the problems of material management in prototype development environment are analyzed. It is observed in the current system that, material management is being carried out manually in a traditional way. Data is collected and documented both as hardcopy and softcopy during these activities, however there is insufficient integration of data coming from different divisions and stores. In addition to that, information transactions between divisions and stores inhibit the activities, causing time planning for production become ineffective. There exists a

high documentation effort. Different documents are being prepared in different formats in different divisions and it is difficult to integrate these data. Besides, inconsistencies are observed among data in different documents. These problems are mainly caused by the nonexistence of an appropriate information system.

The developed information system aims to integrate information from different divisions of the organization and use this information for material management and time planning for production. A multi-level BOM is used as a common structure to access and integrate inventory and configuration information.

The developed system can provide the following benefits:

- The system can handle numerous prototypes and components variation in the system development process. This cannot be achieved using conventional ERP packages which are mainly developed for make-to-stock organizations.
- The system can list the required components, required quantities of these components, required time for acquisition of lacking components and time plan for production.
- Inventory and configuration management information is stored in a well organized way. As a result of this, consistent information among stores and divisions can be achieved.
- Since data in different files and formats is integrated in an organized database, the developed system decreases the documentation effort dramatically in the organization and provides processing of the data stored in the system.
- The major change between the previous system and the designed system is the way of handling BOM versions. While all versions of BOM were stored in different files previously, only the last version of BOM with all change information is stored in the designed database. Using the change information, all the previous versions of BOM are dynamically accessible. This decreased the stored data dramatically and gives opportunity to track the changes on any BOM without any effort.
- The developed system decreases repetitive work which is carried out manually in the current system. Components in the same production batch can be defined and

used as a whole or one by one according to the choice of the operators. This can decrease the required time for these operations considerably.

- The designed system checks all component records in the designed database before entering a new component to the stores. In this way, an item is created only once. Data can be stored in this way consistently.

On the other hand, there are some drawbacks of the developed system. The system has currently no interface with other information systems used in the institute. Besides, being designed for the specific purposes of TÜBİTAK-SAGE, the system is not a general material management system that can be applicable for all type of organizations. It is designed for the needs of an engineer-to-order organization, and some modifications should be made in order to be used in other type of organizations.

In the thesis, a deterministic model is used for time planning for production. As a future work, the model can be extended to handle probabilistic cases such as the best, the moderate and worst cases for durations of material acquisitions, processes and assembly times.

Documents themselves are not included into the designed system. Only information about documents, like document number, is used for production monitoring and planning. Documents can be integrated into the system as a future work.

The BOM structure can be used for ordering items with modification in the developed system. Hence, a procurement module can be added into the system as a future work. In that case, the integrated BOM does not only present the lacking items in the system but it also presents from where the lacking items can be obtained.

REFERENCES

1. Aydın, A. O., Güngör, A., “Effective relational database approach to represent bills-of-materials”, *International Journal of Production Research*, Vol. 43, No. 6, 2005, pp. 1143–1170.
2. Bertrand, J.W.M., Muntslag, D.R., “Production control in engineer-to-order firms”, *International Journal of Production Economics*, Vol. 30-31, 1993, pp. 3-22.
3. Construx, “Detailed Change Control Procedure”, <http://www.construx.com/survivalguide/detailedchangeproc.htm>, July 2006.
4. Du, J., Jiao, Y.-Y., Jiao, J., “Integrated BOM and routing generator for variety synchronization in assembly-to-order production”, *Journal of Manufacturing Technology Management*, Vol. 16, No. 2, 2005, pp. 233-243.
5. Guoli, J., Daxin, G., Tsui, F., “Analysis and implementation of the BOM of a tree-type structure in MRPII”, *Journal of Materials Processing Technology*, Vol. 139, 2003, pp. 535–538.
6. Hass, A. M., *Configuration Management and Principles and Practice*, Glenn Hass First Printing, 2003.
7. Hastings, N.A.J, Yeh, C.-H., “Bill of Manufacture”, *Production and Inventory Management Journal*, Vol. 33, No. 4, 1992, pp. 27–31.
8. Jiao, J., Tseng, M. M., Ma, Q., Zou, Y., “Generic Bill-of-Materials-and-Operations for High-Variety Production Management”, *Concurrent Engineering: Research and Applications*, Vol. 8, No. 4, 2000, pp. 297-322.

9. Kockler, F., Withers, T., Poodiack, J., Gierman, M., System Engineering Management Guide, Defense System Management Collage, Washington, 1990.
10. Kossiakoff, A., Sweet, W. N., System Engineering Principles and Practice, A John Wiley & Sons, inc., Publication 2003
11. Little, D., Rollins, R., Peck, M., Porter, J. K., “Integrated planning and scheduling in the engineer-to-order sector”, International Journal of Computer Integrated Manufacturing, Vol. 13, No. 6, 2000, pp. 545–554.
12. Olsen, K.A., Sætre, P., Thorstenson, A., “A generic bill of materials based on a programming language notion”, in Norsk Informatikk Konferanse (Tapir, Oslo), 1995.
13. Olsen, K.A., Sætre, P., “A visual product constructor for Engineer-to-Order environments”, in 12th International Working Seminar on Production Economics, Igls, Austria, and Norsk Informatikk Konferanse Tromsø, 2001.
14. TÜBİTAK, “Savunma Havacılık ve Uzay Sanayii Paneli Ön Raporu” <http://vizyon2023.tubitak.gov.tr/teknolojiongorusu/paneller/savunmauzayvehavacilik/raporlar/raporshu.pdf>, September 2006
15. Tatsiopoulos, I.P., “On the unification of bills of materials and routings”, Computers in Industry, 1996, pp. 293-304.
16. Veen, E. A. VanWortman, J. C., “Generic bills of material in assemble-to-order manufacturing”, Int. J. Prod. Res., Vol. 25, No. 11, 1987, pp. 1645-1658.
17. Vegetti, M., Henning, G.P., Leone, H.P., “An Object-Oriented Model for Complex Bills of Materials in Process Industries”, Brazilian Journal of Chemical Engineering, Vol. 19, No. 04, 2002, pp. 491- 497.

18. Yeh, C.-H., "Production data modelling: an integrated approach", International Journal of Operations & Production Management, Vol. 15, No. 8, 1995, pp. 52-62.
19. Wikipedia, "Database Normalization", http://en.wikipedia.org/wiki/Database_normalization, December 2006.

APPENDIX A

PROCESS LIST FORM

A sample process form is given in Figure 20.

ÜRÜN İZLEME FORMU																	
BELGE NO : 98F-004619 GÜNÇ. NO : 1 / 2 GÜNÇ. TRH. : 28.09.2003 SAYFA NO :																	
PROJE ADI A projesi		PROJE NO 000001		ÜRÜN ADI Alt Sistem 1													
Sıra No.		PARÇA No.		KARŞI - SERİ No.		İŞ EMİRİ No/ OLUR No		İŞİN TANIMI		İŞ BAŞLAMA TARİHİ		İŞ BİTİŞ TARİHİ		NİTELİK DENET. SONUCU		AÇIKLAMA	
1		130-002401-X1		001-0000		—		İş emiriz bitirilenmiştir.		05.01.2004		26.01.2004		Z		24 adetten oluşmaktadır, UKL'isi M. ÇORUH tarafından hazırlandı.	
2		130-002401-X1		001-0009		—		001 nolu katlıdan alınan bütün		11.02.2004		11.02.2004		Z			
3		130-002401-X1		001-0010		—		001 nolu katlıdan alınan bütün		17.02.2004		17.02.2004		Z			
4		130-002401-05															
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	


Hazırlayan (İmza/Tamimzai): _____

Figure 20 Sample Process List

APPENDIX B

QUALITY CONTROL FORM

The first page of a quality control form is given in Figure 21. The second page of this form is given in Figure 22.

	SAVUNMA SANAYİ ARAŞTIRMA VE GELİŞTİRME ENSTİTÜSÜ	NİTELİK DENETLEME FORMU	BELGE NO : 991-003078 GÜNC. NO : 7 GÜNC. TRH. : 17.11.2004 SAYFA NO : 1/2	
İŞ EMRİ NO:		KAYIT NO:	GİZLİLİK DERECE:	
M/B/AÜ/Ü ADI:		ÜRETİCİ / TEDARİKÇİ ADI:		
ADET/MİKTAR ¹ :		KAFİLE NO ² :		
KAYNAK BELGE ADI		SERİ NO:	KAYNAK BELGE NO	
			GÜNCELLEME NO	
NİTELİK DENETLEME TİPİ	MND <input type="checkbox"/>	BND <input type="checkbox"/>	KND <input type="checkbox"/> END <input type="checkbox"/> TND <input type="checkbox"/>	
DEĞERLENDİRME TABLOSU				
Sıra No	Ölçüm A/G/A Kodu	Olması Gereken	Seri No ³ : Bulgu / Ölçülen	K/Ü/Ö/Y/
1.				
2.				
3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				
11.				
12.				
13.				
14.				
15.				
16.				
17.				
18.				
19.				
20.				
21.				
22.				
23.				
24.				
25.				
NOTLAR				
DEĞERLENDİRME SONUCU				
<input type="checkbox"/> Kabul		Seri No:		
<input type="checkbox"/> Uygunsuz		Seri No:		
NİTELİK DENETLEME İŞLEMİNİ YAPAN (Tarih/Sıra/İmza)		ALTYAPI SORUMLUSU ⁴ (Tarih/Sıra/İmza)		SORUMLU KISIM AMİRİ (Tarih/Sıra/İmza)

1-Nitelik denetleme işleminin kapsadığı M/B/AÜ/Ü'nün toplam adedi/miktarı girilecektir. Örneklerle yapıyor ise "...*..." adetlik kafiyeyle temsil seçilen "...*..." adet numune şeklinde açıklama yapılacaktır.
2-Form sadece bir kafiye için doldurulacaktır.
3-Nitelik Denetleme Birimi tarafından kayıt numarası verildi ise doldurulmayacaktır.
4- Akısi belirtilmedikçe değerlendirme tablosunda yer alan kafiye seri numaraları sırasıyla soldan sağa ve yukarıdan aşağıya doğru artacaktır. Kafiye seri numaralarının tabloda belirtilmemesi durumunda, bu kurala göre değerlendirme yapılacaktır.

Figure 21 Sample Quality Control Form (Page 1)



	SAYINMMA SANAYİİ ARAŞTIRMA VE GELİŞTİRME ENSTİTÜSÜ	NİTELİK DENETLEME FORMU		BELGE NO : 991-003078 GÜNC. NO : 7 GÜNC. TRH. : 17.11.2004 SAYFA NO : 2/2
	İŞ EMRİ NO:		KAYIT NO:	GİZLİLİK DERECESESİ:
	ÜRÜN DEĞERLENDİRME KURULU KARARI⁶			
	1.			
ÜRÜN DEĞERLENDİRME KURULU KARAR ONAYI				
DEĞERLENDİREN / (Tarih/İsmin/İmza) (Tasarımcı/Kullanıcı)		DEĞERLENDİREN / (Tarih/İsmin/İmza) (İş Paralel Lideri/ AP Yürütücüsü)	DEĞERLENDİREN / (Tarih/İsmin/İmza) (Proje Yürütücüsü)	
KONFIGÜRASYON KONTROL KURULU KARARI				
SI No:	DÖ No:		KS Onayı: (Tarih/İsmin/İmza)	
KARAR ÖZET TABLOSU				
Şartlı Kabul Edilen M/B/AU/Ü Seri No:		Şartlı Kabul Edilen M/B/AU/Ü Sayısı:		
Değerli Hurda Olarak Ayrılan M/B/AU/Ü Seri No:		Değerli Hurda Olarak Ayrılan M/B/AU/Ü Sayısı:		
Hurda Olarak Ayrılan M/B/AU/Ü Seri No:		Hurda Olarak Ayrılan M/B/AU/Ü Sayısı:		
Kabul Durumuna Gelen M/B/AU/Ü Seri No ⁶ :		Kabul Durumuna Gelen M/B/AU/Ü Sayısı:		
Düzeltilme İşlemi Gerektiren M/B/AU/Ü Seri No:		Düzeltilme İşlemi Uygulanan M/B/AU/Ü Sayısı:		
Düzeltilme İşlemi İş Emri Numarası:				
UYGUNSUZLUK KAPANMA ONAYI				
Kısmi Amin/Altıyapı Sorumlusu: (Tarih/İsmin/İmza)				

Figure 22 Sample Quality Control Form (Page 2)

APPENDIX C

INVENTORY LIST FORM

An inventory list form is given in Figure 23.

	SAVUNMA SANAYİİ ARAŞTIRMA VE GELİŞTİRME ENSTİTÜSÜ DEPOSU ENVANTERİ	BELGE NO : 991-002647 GÜNC. NO : 3 GÜNC. TRH. : 21.03.2003 SAYFA NO : 1/1
---	--	-------------------------------	--

#	M/B/AÜ/Ü KODU	M/B/AÜ/Ü ADI	MIKTAR	PROJE ADI	YERLEŞİM ALANI	NOT1*	NOT2*
1.							
2.							
3.							
4.							
5.							
6.							
7.							
8.							
9.							
10.							
11.							
12.							
13.							
14.							
15.							
16.							
17.							
18.							
19.							
20.							
21.							

* Depolamanın kendi ihtiyaçlarına uygun olarak Depo Sorumluları tarafından düzenlenecektir (Örnek: Mekanik Ham Malzeme Deposu İçin Nitelik Denetleme İş Emri Numarasının girildiği bir sütun olabilir).

HAZIRLAYAN (Unvan/İmza/Tarih)	ONAYLAYAN (Unvan/İmza/Tarih)
-------------------------------	------------------------------

Figure 23 Sample Inventory List

APPENDIX D

KNAL FORM

A sample KNAL form is given in Figure 24.

[illegible]

Figure 24 Sample KNAL

APPENDIX E

RELATIONSHIP DIAGRAM

The relationship diagram of the designed database, which is given by Microsoft Access, is presented in Figure 25.

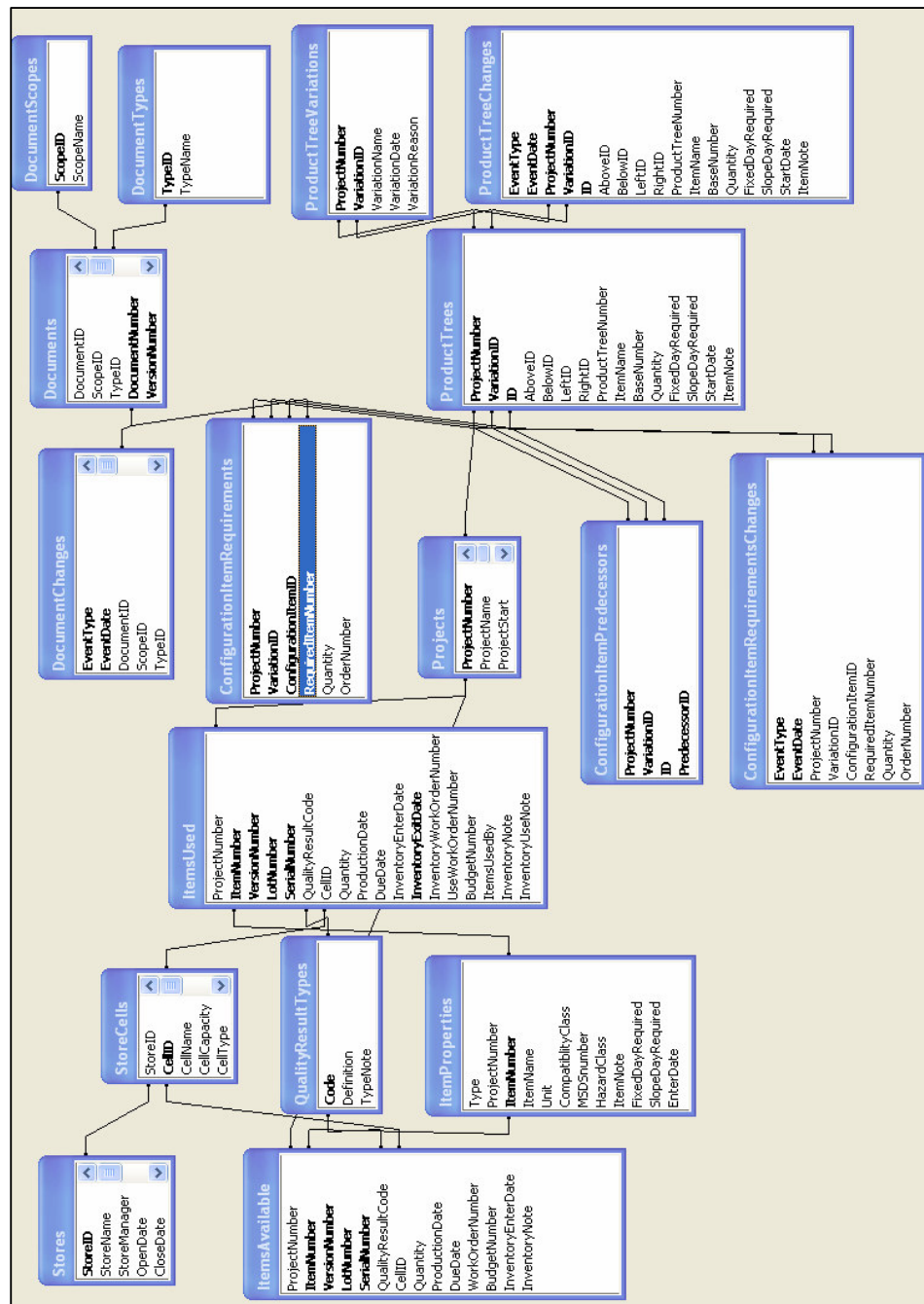


Figure 25 Relationship Diagram of the Designed Database

APPENDIX F

USER INTERFACES OF THE DEVELOPED SYSTEM

The four modules are accessible through a main menu. The main menu is presented in Figure 26.

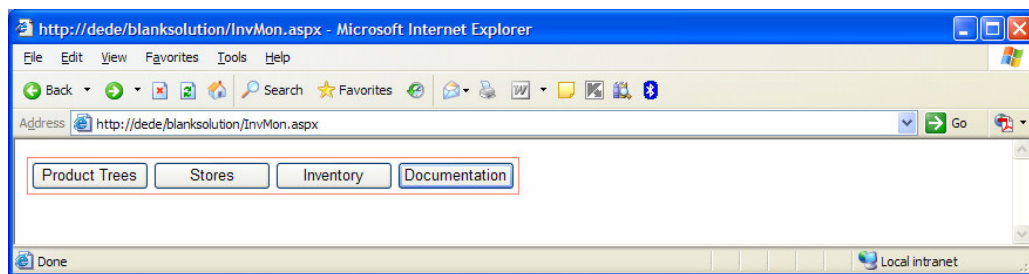


Figure 26 The Main Menu

By clicking “Documentation” button in the main menu, the main page of the documentation module is going to be opened. This page is given in Figure 27.

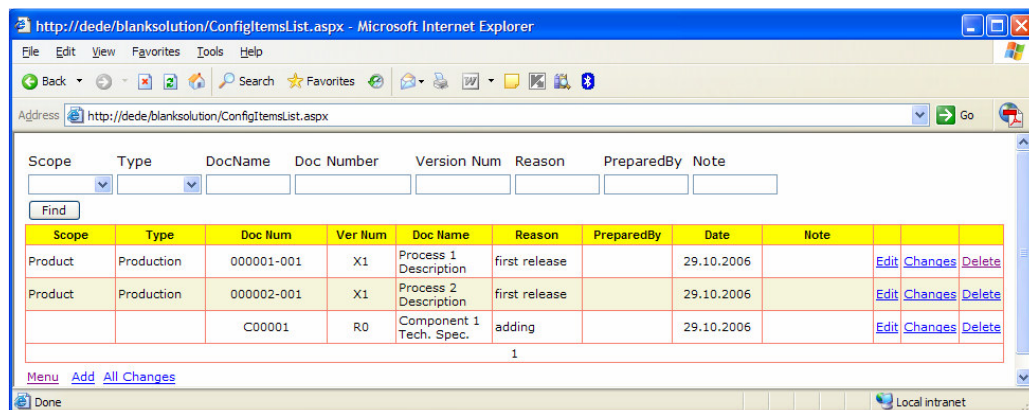


Figure 27 The Main Page of the Documentation Module

If the user clicks “Add” link in the main page of the documentation module, the web browser will direct the user to a page to enter a new document. This page is given in Figure 28.

Figure 28 Document Add

When the user clicks “Changes” link which is located in each document row, the changes related to a document is going to be displayed. The page is presented in Figure 29.

EventType	EventDate	ID	Scope	Type	DocNum	VerNum	DocName	Reason	PreparedBy	Date	Note
Add	29.10.2006 20:37:49	3	Product	Production	000001-001	X1	Process 1 Description	first release		29.10.2006 20:37:49	

Configuration Items List

Figure 29 Document Changes

It is necessary to click inventory button on the main menu for opening inventory module. The main menu of this module is given in Figure 30.

The screenshot shows a web browser window titled "http://dede/blanksolution/InvItemProList.aspx - Microsoft Internet Explorer". The address bar shows "http://dede/blanksolution/InvItemProList.aspx". The main content area contains a search form with fields for Type, Project, Item Number, Item Name, Compatibility Class, Hazard Class, and Item Note. Below the search form is a table with the following data:

Type	Project	Item Num	Item Name	Unit	CompClass	MSDS Num	HazardClass	Note	DayRequired	EnterDate	Item	Item
Process	Project1	000001-001	Process 1	piece					1+6*Q	29.10.2006	Select	Edit
Component	Project1	C00001	Component 1	piece					4+3*Q	29.10.2006	Select	Edit

Below the table are buttons for "Available Items", "Used Items", "Add Items", "Use Items", and "Delete Property". At the bottom left is a "Menu Add" link. The status bar at the bottom shows "Done" and "Local intranet".

Figure 30 The Main Menu of the Inventory Module

By clicking “Add” link in the main menu of the inventory module, a page for defining new inventory item is going to be opened. This page is given in Figure 31.

The screenshot shows a web browser window titled "http://dede/blanksolution/InvItemProAdd.aspx - Microsoft Internet Explorer". The address bar shows "http://dede/blanksolution/InvItemProAdd.aspx". The main content area contains a form for defining new inventory item properties. The form has the following fields:

- Type: Process (dropdown menu)
- Project Name: (dropdown menu)
- Item Number: (text input)
- Item Name: (text input)
- Unit: (text input)
- Compatibility Class: (text input)
- MSDS number: (text input)
- Hazard Class: (text input)
- Day Required: (text input)
- Item Note: (text input)

At the bottom of the form are two buttons: "Add Item Prop." and "Cancel". The status bar at the bottom shows "Done" and "Local intranet".

Figure 31 Defining Inventory Item Properties

After item inventory properties are defined, items can be entered into a store via this module. To enter new items to the stores, “Add Items” button on the main menu of the inventory module must be clicked. This page for entering items to stores is given in Figure 32.

Item Number C00001
Item Name Component 1
Version Number
Project Project1
Lot Number
Serial Number
QualityResult Değerli Hurda
Store
Cell
Quantity/LotSerial piece
Inventory Enter Date 05.01.2007 08:45:10
Production Date
Due Date
Work Order Number
Budget Number
Item Note
Add Parts Cancel

Figure 32 The Page for Entering Items to Stores

After items are entered into the defined stores, the available item list for a specific item can be accessed by clicking “Available Items” button on the main menu of the inventory module. The items available list page is displayed in Figure 33.

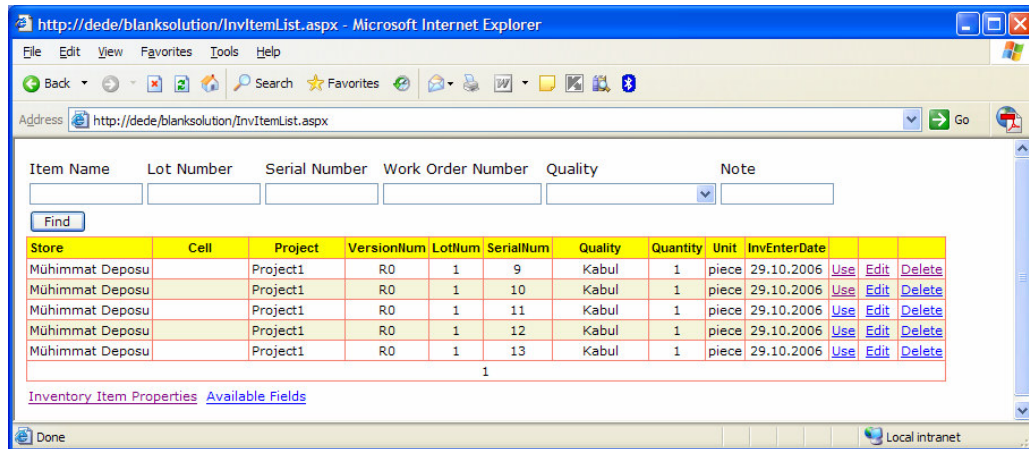


Figure 33 Items Available List

After items entered into the stores, items in the store can be used by clicking “Use” links in the Items Available List. The page for using items in the stores is displayed in Figure 34.

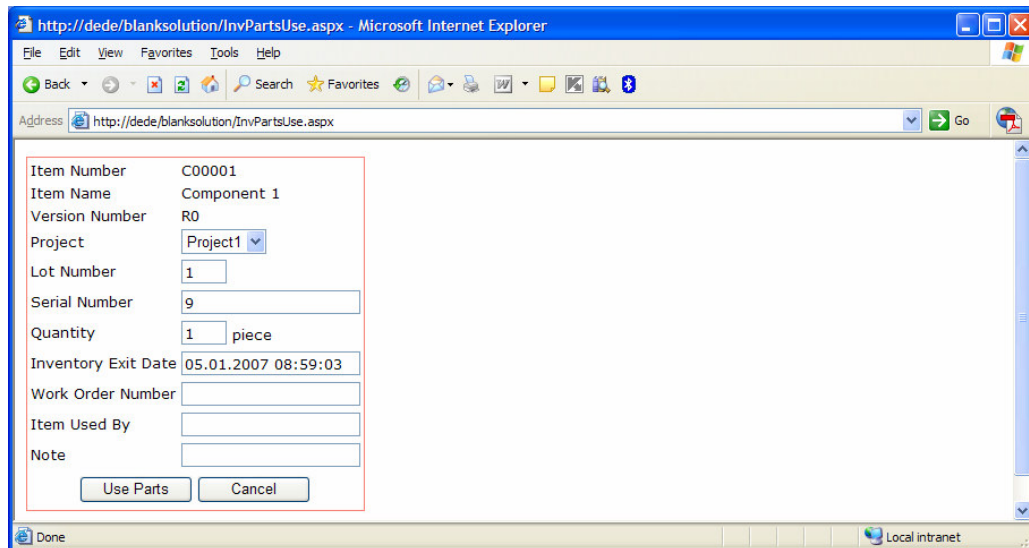


Figure 34 Use Item Operations

The user can also list the used item from stores. For this, “Used Items” button on the main page of the inventory module must be clicked. The page for used items list is given in Figure 35.

Item Name Lot Number Serial Number Work Order Number Quality Note

Find

Store	Cell	Project	VerNum	LotNum	SerialNum	Quality	Quantity	Unit	InventoryEnter
Mühimmat Deposu		Project1	R0	1	1	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	2	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	3	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	4	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	5	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	6	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	7	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	8	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	14	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	15	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	16	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	17	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	18	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	19	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	20	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	21	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	22	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	23	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	24	Kabul	1	piece	29.10.2006
Mühimmat Deposu		Project1	R0	1	25	Kabul	1	piece	29.10.2006

1 2

[Inventory Item Properties](#) [Available Fields](#)

Figure 35 Items Used List

By clicking “Stores” button on the main menu of the developed system, the main menu of the storage module can be opened. The main menu for the storage module is given in Figure 36. From this page, available and used items in one store can be accessed. For this, “Available Items” and “Used Items” buttons can be used.

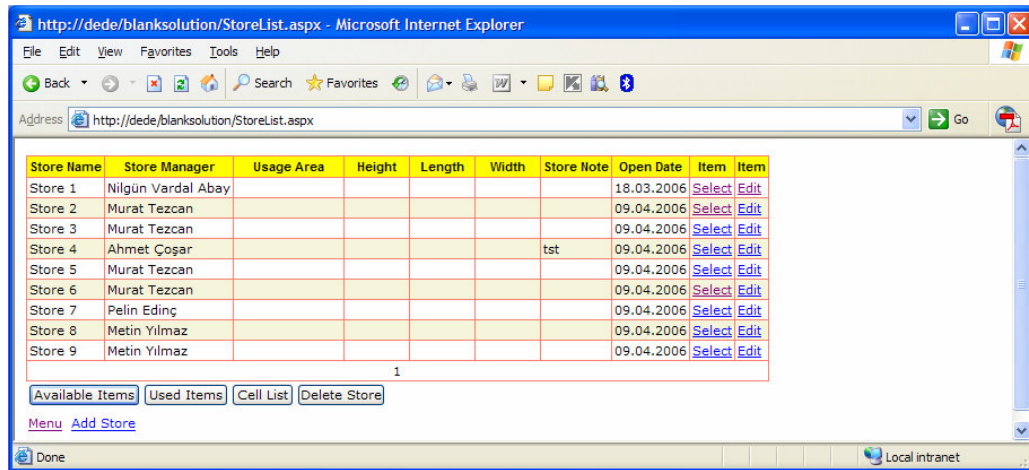


Figure 36 The Main Menu of the Storage Module

To define a new store, “Add Store” link on the main menu of the storage module can be clicked. This will direct the user to store definition page which is presented in Figure 37.

Store Name

Store Manager

Max Usage Area

Open Date

Close Date

Note

Height

Length

Width

Figure 37 Store Definition

In each store, there are different cell places to put items. To list cell places in a store, it is necessary to select a store and click “Cell List” button on the main menu of the storage module. This page is given Figure 38. From this page, available and used items lists are accessible.

Store Name: Store 1

Cell Name	Cell Type	Cell Capacity	Height	Length	Width	Cell Note	Open Date	Item	Item
Diğer Alanlar							18.03.2006	Select	Edit
E							18.03.2006	Select	Edit
F-3-3							18.03.2006	Select	Edit
F-4-3							18.03.2006	Select	Edit
F-5-3							18.03.2006	Select	Edit
F-8-2							18.03.2006	Select	Edit
F-8-3							18.03.2006	Select	Edit
F-9-3							18.03.2006	Select	Edit
Forklift Kullanım Alanı							18.03.2006	Select	Edit
G-1							18.03.2006	Select	Edit
H							18.03.2006	Select	Edit

1

[Available Items](#) [Used Items](#) [Delete Cell](#)

[Store List](#) [Add Store Cell](#)

Figure 38 Cell Lists

To define new cell places in a store, “Add Store Cell” link must be clicked. This link directs to user to Store Cell Definition page which is presented in Figure 39. From this page, available and used items lists are accessible.

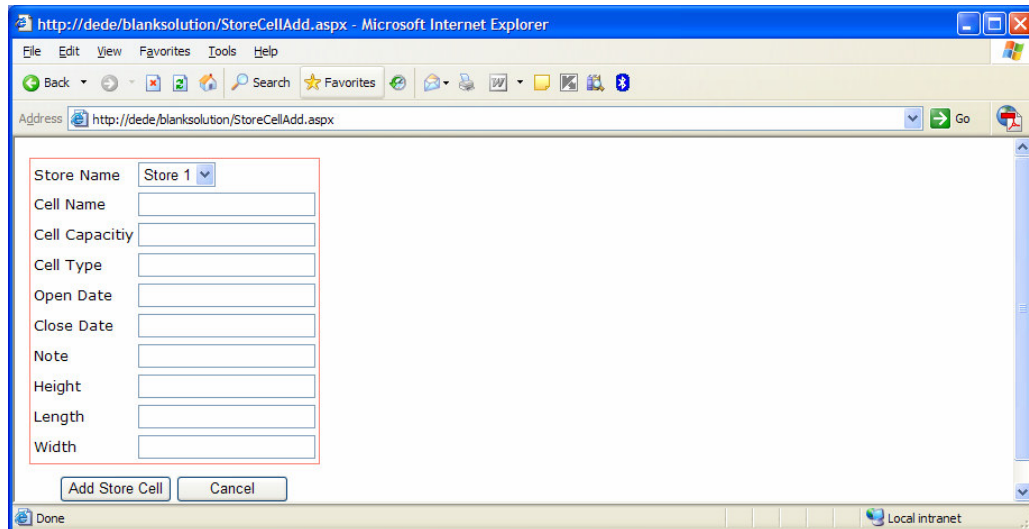


Figure 39 Store Cell Definition

The main menu of the product tree module can be opened by clicking “Product Trees” button on the main menu of the developed system. This page is given in Figure 40.

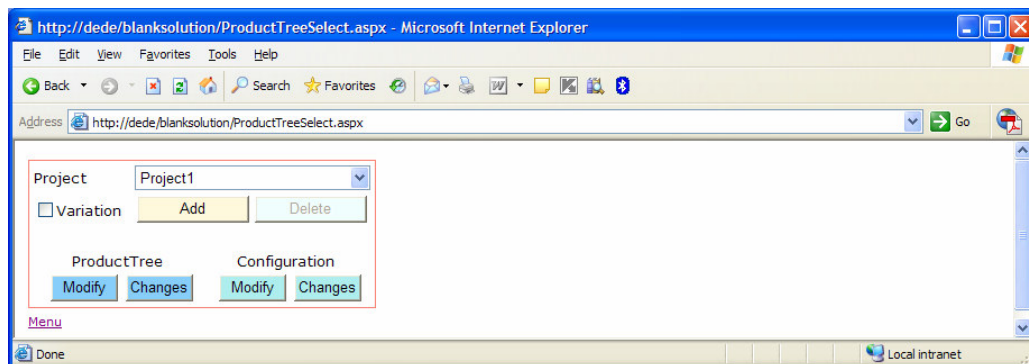


Figure 40 The Main Menu of the Product Tree Module

The main product trees and variants of the main product trees can be modified by clicking “Modify” button on the main of the product tree. The page for modifying product trees is given in Figure 41.

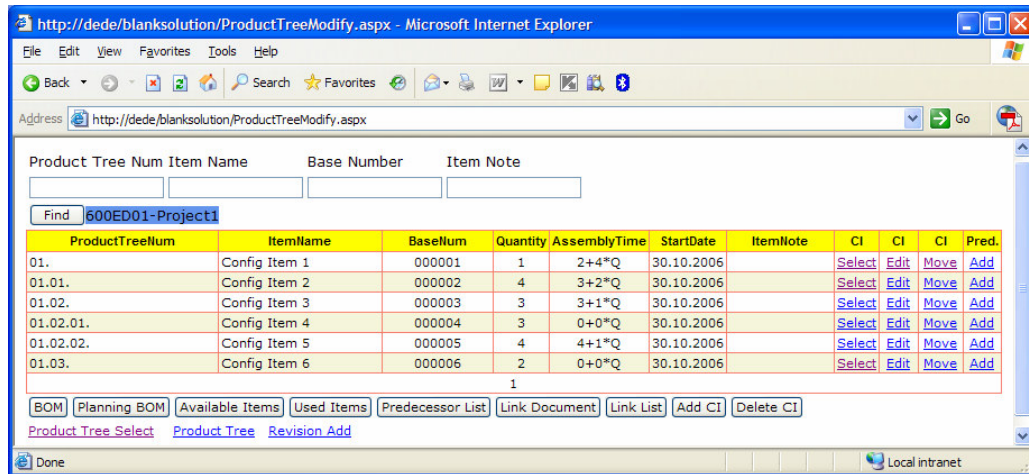


Figure 41 The Page for Modifying Product Trees

To add a new configuration item to a product tree, it is necessary to select the parent item and click to “Add CI” button. This will direct the user to CI defining page which is given in Figure 42.

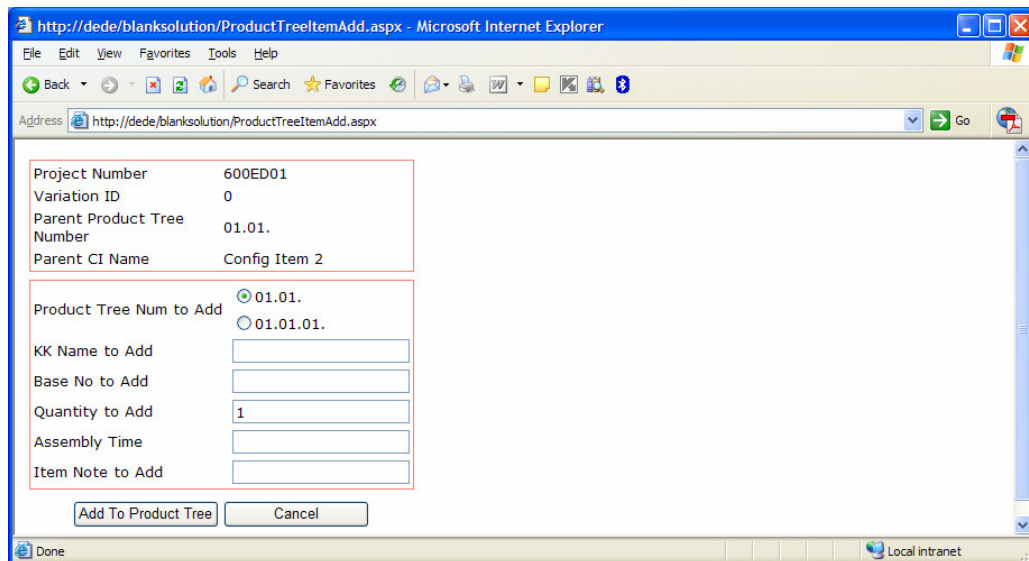


Figure 42 The Page for Defining CIs

To link a document to a CI, “Link Document” link can be used. When the user clicks this link, link document page is going to be open. This page is given in Figure 43.

Project 600ED01
 VariationID 0
 Required Item Number
 Quantity
 Add Date
 Link Item Cancel

Figure 43 Link Document Page

The list of processes and components required by a CI can be accessed by clicking “Link List” button on the page for modifying product tree. This page is displayed in Figure 44.

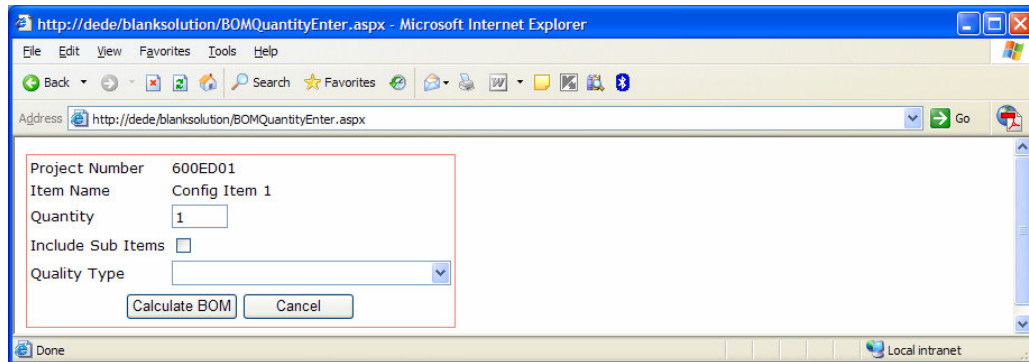
Order	DocScope	DocType	Quantity	Unit	Doc/Item Num	VersionNum	Doc/Item Name	ItemType	DayRequired	Item	Item Move	Move
0			1	piece	C00001	R0	Component 1	Component	4+3*Q	Select	Edit	Up
1	Product	Production	1		000002-001	X1	Process 2 Description	Document		Select	Edit	Up

Available Items Used Items Changes Unlink

Product Tree Modify Link Config Item Item Config Changes

Figure 44 Link List of a CI

To calculate material requirements for a CI, it is necessary to select CI on the page for modifying product tree and click “BOM” button. This will direct the user to a page for entering required quantity for CI. This page is displayed in Figure 45.



The screenshot shows a web browser window titled "http://dede/blanksolution/BOMQuantityEnter.aspx - Microsoft Internet Explorer". The address bar shows the URL "http://dede/blanksolution/BOMQuantityEnter.aspx". The main content area contains a form with the following fields:

- Project Number: 600ED01
- Item Name: Config Item 1
- Quantity: 1
- Include Sub Items: ☐
- Quality Type: [dropdown menu]

Below the form are two buttons: "Calculate BOM" and "Cancel". The status bar at the bottom shows "Done" and "Local intranet".

Figure 45 Entering Required Quantity for a CI

After entering required quantity for the CI is entered, material requirements for the CI can be calculated by clicking “Calculate BOM” button in Figure 45. This will direct to user to material requirement list which is given in Figure 46.

To get time planning for CI production, it is necessary to select CI on the page for modifying product tree and click “Planning BOM” button. This will direct the user to a page for entering required quantity for CI. This page is displayed in Figure 47.

Item Number Version Number Item Name Document Name Lack (>) Excess(>)

Find

ItemNumber	VersionNumber	ItemName	DocumentName	Needed	Available	Lack	Excess	Unit	DayRequired
000001-001	X1	Process 1	Process 1 Description	1	0	1	0	piece	7
000002-001	X1	Process 2	Process 2 Description	1	4	0	3	piece	0
000002-002	X1	Process 3	Process 3 Description	2	1	1	0	piece	9
000002-003	X1	Process 4	Process 4 Description	4	2	2	0	piece	8
000003-001	X1	Process 5	Process 5 Description	3	1	2	0	piece	8
000004-001	X1	Process 6	Process 6 Description	9	5	4	0	piece	13
000004-002	X1	Process 7	Process 7 Description	9	0	9	0	piece	31
000005-001	R0	Process 8	Process 8 Description	12	0	12	0	piece	42
000006-001	X1	Process 9	Process 9 Description	2	10	0	8	piece	0
C00001	R0	Component 1	Component 1 Technical Specification	40	30	10	0	piece	34
C00002	R0	Component 2	Component 2 Technical Specification	48	40	8	0	piece	30
C00003	R0	Component 3	Component 3 Technical Specification	2	12	0	10	piece	0

Product Tree Modify

Figure 46 Material Requirements List

Project Number 600ED01
Item Name Config Item 1
Quantity 1
Quality Type [dropdown]
Calculate Planning BOM Cancel

Figure 47 Entering Quantity For Planning BOM

After entering required quantity for the CI is entered, time planning for production of aCI can be calculated by clicking “Calculate Planning BOM” button in Figure 47. This will direct to user to time planning for production list which is given in Figure 48.

ItemName Lack (>) Excess (>)

ProductTreeItem	ItemName	BaseItem	Quantity	CD	PD	PES	PEF	PLS	PLF	PS	AD	AES	AEF	ALS	ALF	AS
01.	Config Item 1	000001	1	0	7	0	7	57	64	57	6	64	70	64	70	0
01.01.	Config Item 2	000002	4	34	17	0	17	30	47	30	11	34	45	47	58	13
01.02.	Config Item 3	000003	3	0	8	0	8	50	58	50	6	58	64	58	64	0
01.02.01.	Config Item 4	000004	9	0	44	0	44	14	58	14	0	44	44	58	58	14
01.02.02.	Config Item 5	000005	12	34	42	0	42	0	42	0	16	42	58	42	58	0
01.03.	Config Item 6	000006	2	0	0	0	0	64	64	64	0	44	44	64	64	20
				1												

[Product Tree Modify](#)

Figure 48 Time Planning for a CI production

A product tree variant can be created in two ways. In the first way, the current product tree is used as a basis for variant product tree. To do this, variation checkbox must be selected in the Main Menu of the Product Tree Module, then “Add” button must be clicked. This will direct the user the page which is given in Figure 49. If the user clicks “Add Variation” button in this page, a variant product tree will be created by using the current project product tree. In the second way, previous version of the product tree is used for creating product tree variants. To do this, a time value must be supplied. The version of product tree at the given time is used for creating variant product tree in this case.

Project Number 600ED01
 Variation ID 1
 Variation Name
 Variation Reason
 Variation Date 05.01.2007 10:30:50

Figure 49 Add Variation Page