METHODOLOGY DEVELOPMENT
FOR SMALL AND MEDIUM SIZED ENTERPISE (SME) BASED
VIRTUAL ENTERPRISES

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

BY

BURAK SARI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

JUNE 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 Doctor of Philosophy.

Prof. Dr. Kemal İDER
Head of Department

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

Prof. Dr. Sadık Engin KILIÇ
Supervisor

Examining Committee Members

Prof. Dr. Mustafa İlhan GÖKLER	(METU, ME)	____________________
Prof. Dr. S. Engin KILIÇ	(METU, ME)	____________________
Assoc. Prof. Dr. Tayyar ŞEN	(METU, IE)	____________________
Prof. Dr. Ömer ANLAĞAN	(TÜBİTAK)	____________________
Prof. Dr. Can ÇOĞUN	(GAZİ ÜNV., ME)	____________________

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.

Burak SARI

ABSTRACT

METHODOLOGY DEVELOPMENT
FOR SMALL AND MEDIUM SIZED ENTERPRISE (SME) BASED VIRTUAL ENTERPRISES

SARI, Burak

Ph. D., Department of Mechanical Engineering

Supervisor: Prof. Dr. S. Engin Kılıç

June 2006, 371 pages

This dissertation presents the results of a Ph.D. research entitled as *methodology development for SME based virtual enterprises*. The research addresses the preparation and set up of virtual enterprises and enterprise networks. A virtual enterprise (VE) can be perceived as a customer solution delivery system created by a temporary and re-configurable information and communications technology (ICT) enabled aggregation of competencies.

The main achievements of the research include:

- **Clarification and definition of the concept** for virtual enterprises and enterprise networks including preparation of these.
    - A fast and efficient setup of virtual enterprises can be enabled through the establishment of an enterprise network in which an appropriate type and degree of work preparation can be established prior to the set up of virtual enterprises.
- **Development of a framework and a reference architecture** for virtual enterprises named as Structured Methodology and ICT Reference Architecture respectively.

  - Structured Methodology structures the body of knowledge related to preparation, set up and operation of virtual enterprises and enterprise networks.
  - ICT reference architecture consists of three levels with seven layers to portray in a diagrammatic fashion how different enterprises may exchange and use information between their respective organizations' specific proprietary systems and a central server.
- **Development of a methodology for virtual enterprise** named as Virtual Enterprise Methodology (VEM)
  - VEM consists of a set of guidelines, which systematically describes activities that enterprises should consider in relation to set up and preparation of own enterprise networks with the aim to set up virtual enterprises.
- **Testing and validation** of the developed VEM with the realization of a virtual case study and establishment of a validation platform respectively.
  - Virtual case study demonstrates the application of the developed VE methodology with the illustration of the key activities related to setting up breeding environment, setting up & operating VE and dissolution of VE.
  - The findings in the research can be validated through the various activities as meetings, conferences, presentations and publication of journals.

Keywords: Virtual Enterprise, Enterprise Network, Breeding Environment, ICT, IDEF, UML, Analytic Hierarchy Process, Artificial Neural Network, VEM.

ÖZ

KÜÇÜK VE ORTA ÖLÇEKLİ FİRMALARA (KOBİ) YÖNELİK SANAL FABRİKA METODOLOJİSİ GELİŞTİRİLMESİ

SARI, Burak
Doktora, Makina Mühendisliği Bölümü
Danışman: Prof. Dr. S. Engin Kılıç

Haziran 2006, 371 sayfa

Bu doktora tezi KOBİ'lere yönelik sanal fabrika metodolojisi geliştirilmesi isimli araştırmanın sonuçlarını sunmaktadır. Araştırma, sanal fabrikanın kurulumu ve çalıştırılması ve firma ağı yapısının kurulumu ile ilgili çalışmaları kapsamaktadır. Sanal fabrika, bağımsız şirket veya kurumların çekirdek yeteneklerini bilgi ve iletişim teknolojilerinin (BİT) etkin kullanımı ile biraraya getirerek müşteri taleplerini karşılamak için oluşturduğu geçici ve yeniden yapılandırılabilir bir ağ olarak tanımlanabilir.

Bu çalışmanın ana sonuçları şu şekilde sıralanabilir:

- Sanal fabrika ve firma ağı **kavramlarının tanımlanması**.
    - Sanal fabrikaların hızlı ve verimli bir şekilde kurulumu, kurulum ile ilgili bazı işlerin istenen seviyede önceden hazırlanmasını sağlayan firma ağı ile gerçekleştirilir.
- Sanal fabrikalar için sırasıyla Sanal Fabrika Uygulama Metodolojisi ve BİT Referans Mimarisi isimli bir **sistem yapısı ve referans mimarisinin geliştirilmesi**.
    - Sistem yapısı sanal fabrika ve firma ağlarının hazırlanması, kurulumu ve çalıştırılması ile ilgili bilgi yapısının oluşturulmasını sağlar.

    - BİT referans mimarisi farklı organizasyonlar veya firmalar arasındaki bilgi değişiminin ve kullanımının nasıl sağlanacağını yedi katmanlı üç seviyeden oluşan bir yapıyla açıklar.
- Sanal Fabrika Metodolojisi (SFM) isimli **sanal fabrikalara yönelik bir metodolojinin geliştirilmesi**.
    - SFM firma ağı ve sanal fabrikaların kurulumu, çalıştırılması ve kapanışı ile dikkat edilmesi gereken hususları ve aktiviteleri sistematik bir şekilde anlatan bir yönerge içerir.
- Sanal bir pilot uygulamanın yürütülmesi ve bir onaylama platformunun oluşturulmasıyla geliştirilen SFM'nin **test edilmesi ve doğruluğunun onaylanması.**
    - Sanal pilot uygulaması geliştirilen SFM'nin uygulanabilirliğini filizlenme çevresinin kurulumu, sanal fabrikanın kurulumu, çalıştırılması ve kapanışı ilgili anahtar aktiviteleri adım adım göstererek ispatlar.
    - Araştırma sonucunda ortaya çıkan bulgular çeşitli aktivitelerle doğrulanabilir; toplantılar, konferanslar, sunumlar ve uluslararası dergiler.

Anahtar Kelimeler: Sanal Fabrika, Firma Ağı, Filizlenme Çevresi, BİT, IDEF, UML, Analitik Hiyerarşi Prosesi, Yapay Sinir Ağı, SFM.

to my lovely family

ACKNOWLEGEMENTS

I would like to express my deepest gratitude and appreciation to my supervisor Prof. Dr. S. Engin Kılıç for his guidance, encouragement, and continual personal and academic support.

I would like to express sincere thanks to Prof. Dr. Ömer Anlağan and Assoc. Prof. Dr. Tayyar Şen for their valuable comments and contributions to this work.

Many thanks to my colleagues in the Integrated Manufacturing Technologies Research Group (IMTRG) for the conversations and asides help that made each day more enjoyable. Special thanks to Saleh Amaitik, Yusuf Başıbüyük, and Burak Ay. I extend my thanks and appreciation to Serdar Gürel from Eko Mühendislik and to all others who contributed directly or indirectly, to bring out this thesis successfully.

I would also like to express my gratefulness and appreciation to the Turkish State Planning Agency, The Scientific and Technical Research Council of Turkey and Graduate School of Natural and Applied Sciences of METU for their support throughout this research.

Finally, greatest thanks to my family, without their existence completion of this thesis would not be possible.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
ANN	Artificial Neural Network
ASP	Active Server Pages
B2B	Business-to- Business
B2C	Business-to-Customer
BP	Business Process
C/S	Client/Server
COM	Common Object Model
DMS	Distributed Manufacturing Systems
DNA	Distributed Internet Applications
EDI	Electronic Data Interchange
ERP	Enterprise Resource Planning
ICT	Information and Communication Technology
IDEF	Icam DEFinition
IIS	Internet Information Server
IMTRG	Integrated Manufacturing Technologies Research Group
OKP	One-of-a-kind-production
OOP	Object Oriented Programming
PLC	Product Life Cycle
RM	Reference Model
RQ	Research Question
SCM	Supply Chain Management
SME	Small and Medium size Enterprises
STEP	Standard for the Exchange of Product Model Data (ISO 10303)
SQL	Structured Query Language
UML	Unified Modeling Language
VB	Visual Basic

VBE	Virtual enterprise Breeding Environment
VE	Virtual Enterprise
VEM	Virtual Enterprise Methodology
VMP	Virtual Market Place
WWW	World Wide Web
XML	Extensible Mark-up Language

CHAPTER 1

INTRODUCTION

1.1 Background

Companies of the 21st century are faced with increasing demands from the market. These demands are among other things derived from technological innovations and the corresponding development towards a global marketplace. Specific advantages deriving from operating in a global market appear to be exploitable only by large organizations. Small and medium size enterprises (SMEs), in particular, must find organizational solutions that allow them to cope with global business opportunities without suffering the effects of their limited resources or exposing themselves to the risk of direct investment. An analysis of the specific context of SMEs reveals that many small high tech enterprises have failed owing to their lack of technical and management competencies.

The increasing significance of the so-called ‘new economy’, also referred to as the digital and knowledge economy push towards further concentration on corporate competencies while exploiting and developing these competencies in inter-organizational networks facilitated by advanced information and communication technologies (ICTs).

In this context the notion of the virtual enterprise is receiving increasing attention as a business model addressing these new business challenges. A virtual enterprise (VE) can be perceived as a customer solution delivery system created by a temporary and re-configurable ICT enabled aggregation of competencies. When the customer’s demand has been satisfied, the virtual enterprise is decommissioned. The virtual enterprise is therefore highly agile compared to conventional, more rigid manufacturing chains.

The virtual enterprise is tightly related to the ability to set up an enterprise uniquely tailored for a specific delivery and composed of competencies from independent partners often not well acquainted. Besides, a crucial prerequisite for the existence of virtual enterprise is the time it takes to deliver the solution, and as a part of this, the time it takes to set up the virtual enterprise. It is of no use being able to deliver a 'first class solution' if the time to market is not competitive.

Until till now research into virtual enterprises and related business-to-business initiatives have in general focused on technological solutions and the linking of distributed information systems. However, technological solutions are not sufficient. Several other aspects need to be explored as well, e.g. the risk/uncertainty related to co-operating with new partners (are the partners trustworthy?) in a global environment, the quality of decision making in the partner companies' performance evaluation and the decision support functionality, which addresses the issue such as partner company selection using intelligent algorithms.

As mentioned above the setup of a virtual enterprise is an important aspect for the competitiveness of the virtual enterprise. This implies that the individual enterprises need to prepare for participation in virtual enterprises. The preparation can take place within the individual enterprises as well as coordinated activities between enterprises in a VE-focused enterprise network. Last mentioned can be seen as the breeding ground for the creation of virtual enterprises. Breeding environment establishes mutual agreements among its members on issues such as common standards, procedures, intellectual property rights, and ICT, so that these time-consuming preparations can be significantly shortened when a customer request arises, and a VE is put in place.

1.2 Purpose

In the following, the Research Questions (RQs) of this Ph.D. research is described.

They are stated in a first introductory sense, i.e. as the author of this dissertation perceived the problems when the research work was initiated. The research addresses VE and the questions are structured around three very basic questions: why, what and how as indicated on Figure 1.1.

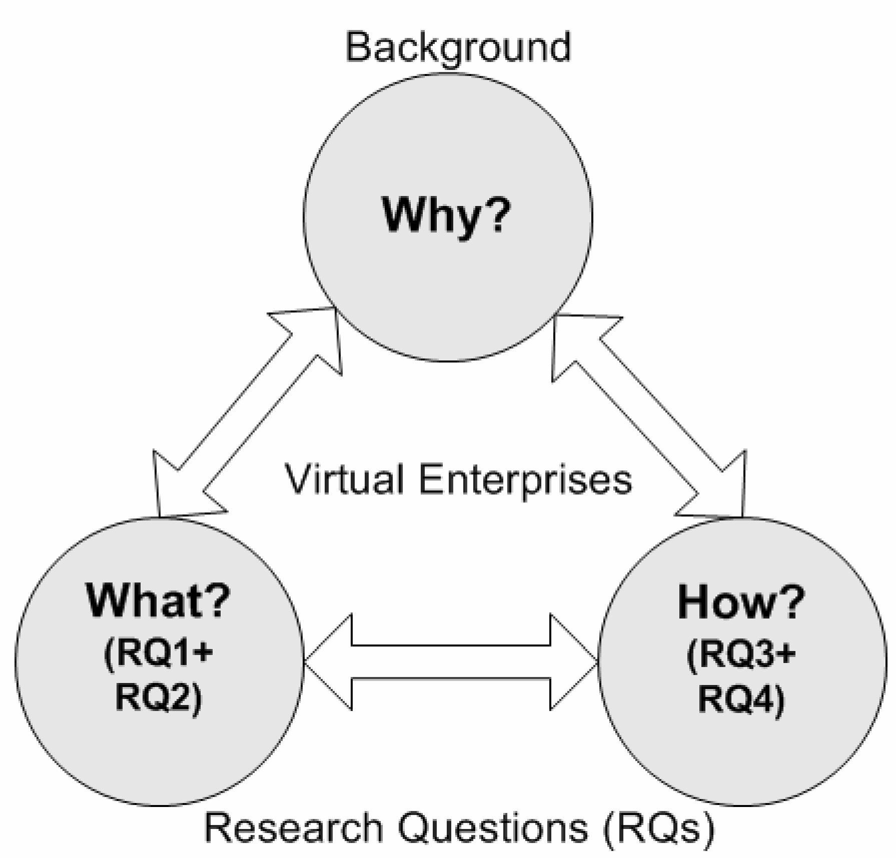


Figure 1.1 The three basic questions addressed in this research

The research aims to answer the following questions:

- **<u>Why</u>** is there a need for new organizational forms such as the virtual enterprise? (i.e., what are the business conditions/drivers?)
    - This question is not a research question in this dissertation as such. It is to be considered as a background question clarifying the need for addressing virtual enterprise. Thus, no new knowledge is expected to be revealed.
- **<u>What</u>** is a virtual enterprise?
    - This will be addressed in the first research question (RQ1), cf. section 1.2.2.

- **<u>What</u>** type of knowledge is needed to realize VEs? (and how could this knowledge be structured?)
    - This will be addressed in the second research question (RQ2), cf. section 1.2.3.
- **<u>How</u>** to prepare, set up and operate VEs? (applying the identified types of knowledge)
    - This will be addressed in the third research question (RQ3), cf. section 1.2.4.
- **<u>How</u>** to test and validate VE methodology?
    - This will be addressed in the fourth research question (RQ4), cf. section 1.2.5.

1.2.1 Background question

Throughout the literature a need for new enterprise forms are described. Traditional ways of organizing enterprises in stable supply chains based on long-term partnerships will no longer be sufficient in today's global environment.

> *"We are at the dawn of a new industrial order… We are taking our first tentative steps into a world where imagination, experimentation, and agility are, if not everything, at least the essential catalysts for wealth creation"* (Hammel, 1999)

One of the business models that are receiving attention is the agile virtual enterprise model, which emerged around a decade ago as a response to this demand (Davidow & Malone, 1992; Goldman et al, 1995; Katzy et al, 1996; Goranson, 1999). As mentioned, the first question of this Ph.D. is not a research question as such; the question is considered as a background question addressing why the virtual enterprise business model is getting this attention. What are the trends facing today's enterprises.

Background Question: Why is the virtual enterprise relevant (what business trends *are making it relevant)?*

Thus, it is not the intention to clarify if there is a need for more agile and dynamic enterprise forms such as virtual enterprises. The research takes for granted that there is a need for new business forms. What the background question aims to clarify is what the drivers are behind dynamic enterprise forms such as the virtual enterprise. This background question will be answered primarily through literature study.

1.2.2 Research question 1 – what is a VE?

In prolongation to clarifying the business conditions triggering the emergence of a new and more agile business model, a concurrent next step will be to clarify what a virtual enterprise is.

In its theoretical form the virtual enterprise seems ideal, i.e. based upon a specific customer need, companies possessing the most suitable competencies will get together and create the solution requested by the customer. However, this is easier said than done, a set of challenges is related to the virtual enterprises business model. One of the bigger challenges is related to the dynamic aspect of a VE such as dealing with new and unknown partners on a more frequent basis. Challenges related to dealing with new partners include:

- Trust issues (is the company trustworthy)
- Cultural differences (e.g., differences in values)
- Language difficulties due to different native languages
- Terminology (different use and understanding of terms)
- ICT interoperability (or lack hereof)

Taking outset in these conditions this research aims to determine a suitable concept for the dynamic inter-enterprise environments by researching the question:

RQ1: What is a virtual enterprise?

At the commencement of this research the author of this dissertation had an initial understanding about what a virtual enterprise is. The aim of the research question is to clarify the virtual enterprise concept based upon a literature review and determine a business concept that suits the situation of industrial enterprises, i.e. the drivers that were identified as a part of the background question outlined above. That is, the research is expected to contribute with a new and more clarified understanding of the virtual enterprise concept. As already indicated this RQ will primarily be addressed through a literature study.

1.2.3 Research question 2 – what framework is needed?

Although the concept of a virtual enterprise is of a more recent date most of the knowledge needed to set up a VE is not all new. The list of requirements for what is relevant in relation to virtual enterprise includes many aspects such as:

- Different types of models with regards to, e.g. procedural and functional models, information models
- Legal aspects
- Organizational issues
- ICT applications
- ICT infrastructure

One of the problems however is to get an overview of what knowledge is needed as well as what is already available and what needs to be developed further. What is lacking is a comprehensive framework that outlines and structures the tasks and thereby the type of knowledge that is needed to facilitate the setup and operation of virtual enterprises. Thus, the second research question can be stated as:

RQ2: What framework is needed to structure the body of knowledge related to preparing, setting up and operating virtual enterprises?

As a result of this question, the research aims to specify a comprehensive framework that can support managers and enterprise engineers to establish an overview of what type of tasks are needed to be considered as well as what type of knowledge is related to the activities.

1.2.4 Research question 3 – how to prepare, set up & operate VEs?

Having clarified the business context of the VE as well as the type of knowledge enterprises need to posses the next step will be to provide guidelines for how to use this knowledge to set up a virtual enterprise. Accordingly, the research question in this research is stated as:

RQ3: How to set up & operate VEs including what to consider when preparing for it?

This research aims to provide guidelines for what needs to be considered in terms of preparing for and setting up and operating virtual enterprises as well as their breeding environment i.e. the enterprise networks.

1.2.5 Research question 4 – how to test & validate VE methodology?

The concluding understanding of a VE methodology should be tested with the realization of virtual case studies. The findings in the research should be validated throughout the research through various activities. Thus, the last research question can be stated as:

RQ4: How to test & validate VE methodology?

One of the aims of this research is to carry out certain test cases in order to test the proposed VE methodology. The validation of the platform for setting up VEs can be done by the following activities; meetings, conferences, presentations and realization of individual application scenarios.

1.3 Outline

In addition to this chapter, the thesis is organized in eight chapters that present in detail modeling & software technologies, the state of art in the area of VEs, the proposed ICT reference architecture and developed structured methodology, VE methodology, and the validation and assessment of the proposed concept. The outline of the dissertation is shown in Figure 1.2.

In this first chapter, the background of the research has been described along with a first description of the research questions addressed in this dissertation.

In chapter 2, modeling technologies, which are widely accepted and used in software projects, are reviewed. Furthermore, advanced software technologies as information systems, database management systems, distributed systems and technologies, and Microsoft .NET framework are reviewed.

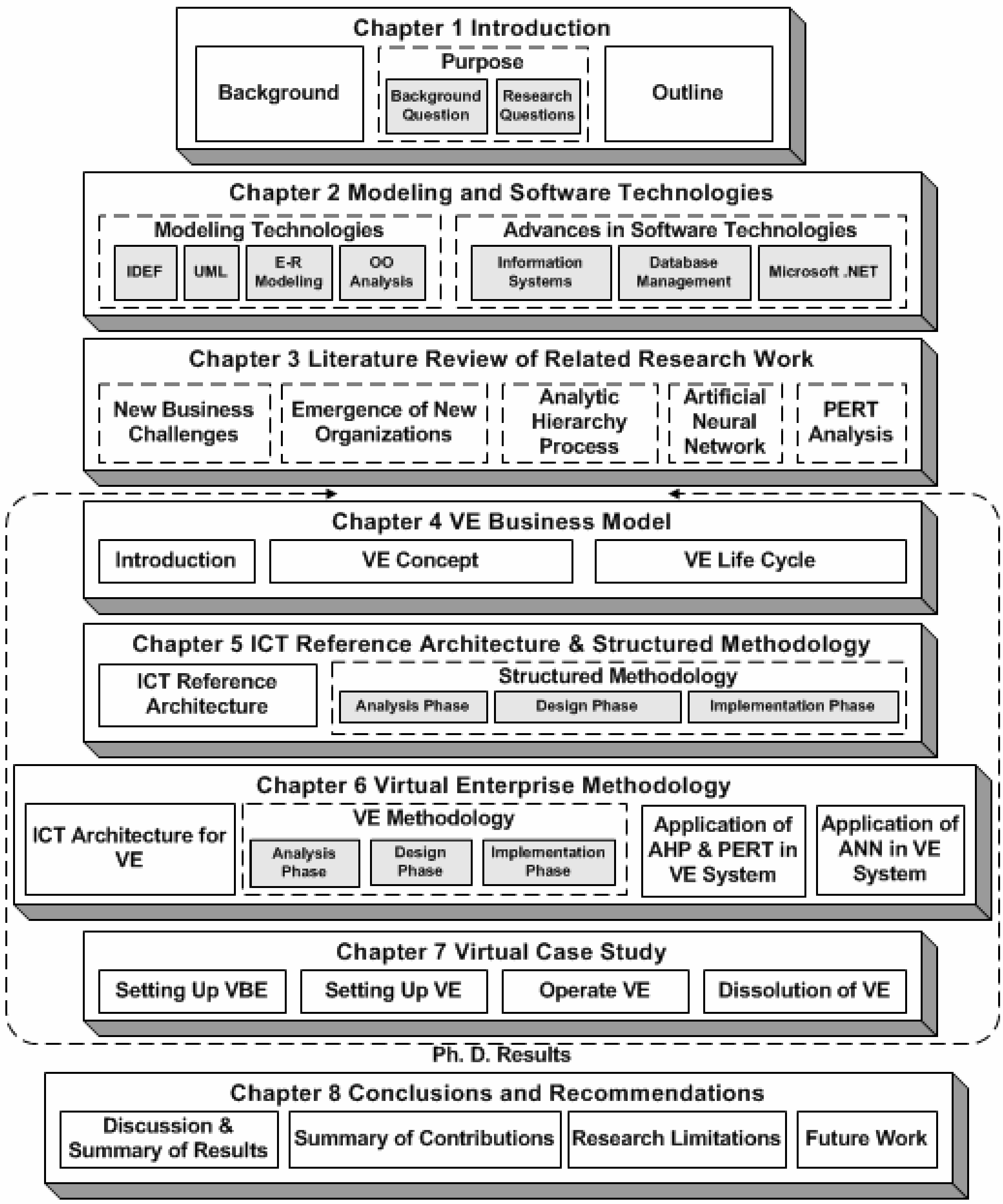


Figure 1.2 Outline of dissertation

In chapter 3, the theoretical elements of this research are described. The five main research streams covered are: new business challenges, new enterprise forms focusing on virtual enterprises, analytic hierarchy process (AHP) to contribute to the selection of the partner companies in the virtual enterprises, artificial neural network (ANN) to assess the performance of the partner companies taking into consideration the disciplinary and quality aspects and finally program evaluation review technique

(PERT) to calculate the probability of completing the project by the given time period.

The results of this Ph.D. are described in chapters 4 to 7. In chapter 4, the VE concept is described as an answer to the RQ1. In chapter 5, the proposed systems framework for virtual enterprise systems and the developed reference ICT architecture are described as a partial answer to RQ2. In chapter 6, the virtual enterprise methodology & ICT architecture are described as an answer to RQ3. In chapter 7, realization of case study including a set of activities to go through when setting up VBE, setting up and managing VE and establishment of a validation platform are discussed.

Chapter 8 summarizes the main research accomplishments of this thesis and describes the anticipated impact and research limitations. It is concluded by identifying directions for future research.

CHAPTER 2

SYSTEM MODELING AND SOFTWARE TECHNOLOGIES

2.1 System Modeling Technologies

2.1.1 IDEF specification

During 1970s, the U.S. Air Force Program for Computer Integrated Manufacturing (ICAM) sought to increase manufacturing productivity through systematic application of computer technology. The ICAM program identified the need for better analysis and communication techniques for people involved in improving manufacturing productivity. As a result, the ICAM program developed a series of techniques known as the ICAM DEFinition (IDEF) techniques, which included the following:

1. IDEF0, used to produce a “functional model”. A functional model is a structured representation of the functions, activities, or processes within the modeled system or subject area.
2. IDEF1, used to produce an “information model”. An information model represents the structure and semantics of information within the modeled system or subject area.
3. IDEF2, used to produce a “dynamics model”. A dynamics model represents the time-varying behavioral characteristics of the modeled system or subject area.

In 1983, the U.S. Air Force Integrated Information Support System program enhanced the IDEF1 information modeling technique to form IDEF1X (IDEF1 Extended), a semantic data modeling technique.

Currently, IDEF0, IDEF1X techniques are widely used in the government, industrial and commercial sectors, supporting modeling efforts for a wide range of enterprises and application domains.

2.1.1.1 IDEF0 methodology

IDEF0 based on SADT^{TM} (Structured Analysis and Design Techniques) developed by Douglas T. Ross and SofTech, Inc (Ross, 1985). In its original form, IDEF0 includes both a definition of graphical modeling language (syntax and semantics) and a description of a comprehensive methodology for developing models (IDEF0, 1993).

IDEF0 may be used to model a wide variety of automated and non-automated systems. For new systems, IDEF0 may be used first to define the requirements and specify the functions and then to design an implementation that meets the requirements and performs the functions. For existing systems, IDEF0 can be used to analyze the functions the system performs and to record the mechanisms (means) by which these are done.

As a function modeling language, IDEF0 has the following characteristics:

1. It is comprehensive and expressive, capable of graphically representing a wide variety of business, manufacturing and other types of enterprise operations to any level of detail.
2. It is coherent and simple language, providing for rigorous and precise expression, and promoting consistency of usage and interpretation.
3. It enhances communication between systems analysts, developers and users through ease of learning and its emphasis on hierarchical exposition of detail.
4. It is well-tested and proven, through many years of use in Air Force and other government development projects and by private industry.
5. It can be generated by a variety of computer graphics tool; numerous commercial products specifically support development and analysis of IDEF0 diagrams and

models.

The IDEF0 models consist of a hierarchy of related diagrams. Each diagram is based on a diagonal row of boxes (normally between three and six boxes on each diagram) connected by a network of arrows. The boxes represent activities, which are described by an active verb phrase contained in the box. Arrows represent the relationship between activities in terms of the information or objects used, produced or required by the activities. Arrows entering the left side of a box are inputs (I) to the activity, arrows entering the top of a box are controls (C) on the activity and arrows leaving the right side of a box are outputs (O) as a result of the activity. Finally a mechanism (M) is a person, system or device associated with carrying out the activity and is shown as an arrow entering the base of the box. Mechanism arrows that point downward are call arrows. Call arrows enable the sharing of detail between models (linking them together) or between portions of the same model. The called box provides detail for the caller box. This arrow structure is termed the ICOM (Input, Control, Output, and Mechanism) structure (Figure 2.1).

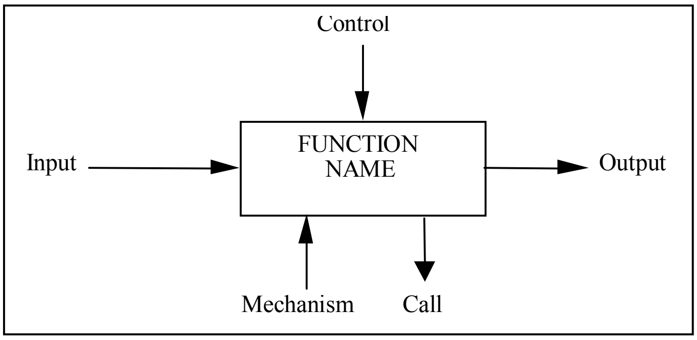


Figure 2.1 IDEF0 function box (IDEF0, 1993)

One of the most important features of IDEF0 as a modeling concept is that it gradually introduces greater and greater levels of detail through the diagram structure comprising the model. In this way, communication is enhanced by providing the reader with a well-bounded topic with a manageable amount of detail to learn from each diagram. Each diagram is referred to by its “node number” that defines where it lies in the hierarchy model (Figure 2.2).

IDEF0 representation and decomposition of functions can be used to present a model of virtual enterprise system. This strategy allows defining the functions and relationships of program modules through a systematic manner early in the design phase. Any output arrow may provide some or all of the input, control, or mechanism data or objects to any other box. An example of a typical IDEF0 activity diagram format with its model relationship and text is shown in Figure 2.3.

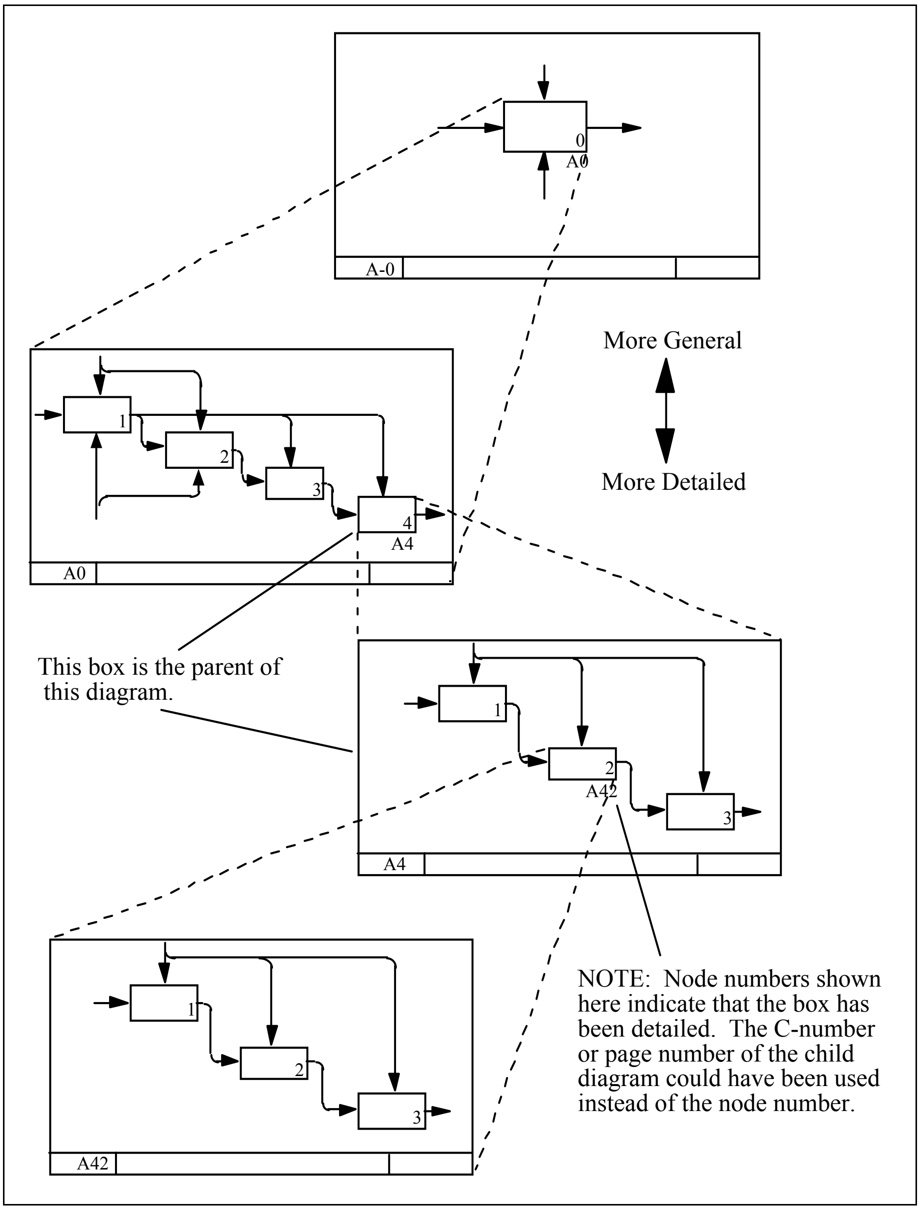


Figure 2.2 IDEF0 hierarchy (IDEF0, 1993)

A node index and node tree provide a conventional way of showing the relationship between all the diagrams in a model and a data dictionary can be used to summarize all sources and destinations of objects, information and data (Colquhoun et al, 1993).

IDEF0 may be combined with the IDEF1X to build the information model as well (Unver, 2000). The following section discusses the IDEF1X model.

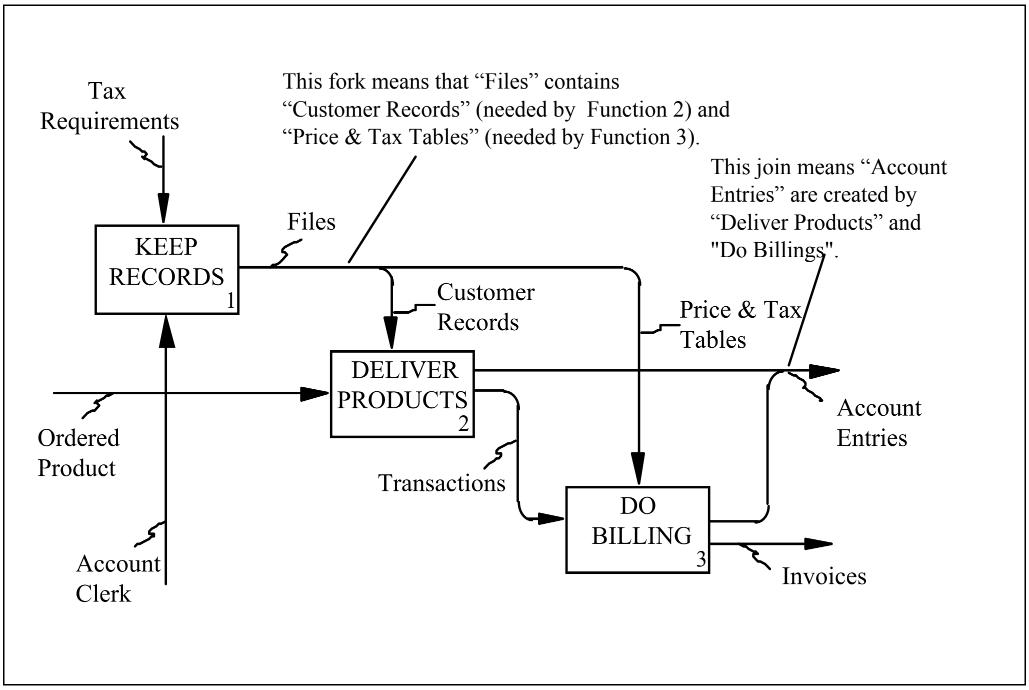


Figure 2.3 Sample IDEF0 diagram (IDEF0, 1993)

2.1.1.2 IDEF1/X methodology

Application within industry had led to the development in 1982 of a Logical Database Design Technique (LDDT) by R. G. Brown of the Database Design Group. The technique was based on the relational model of Dr. E. F. Codd and the entity relationship model of Dr. P. P. S. Chen. It provided multiple levels of models and a set of graphics for representing the conceptual view of information within enterprise. LDDT had a high degree of overlap with IDEF1 features, introduced enhanced semantic and graphical constructs and addressed information modeling enhancement requirements identified under the Integrated Information Support system (I^2S^2)

program. Under the technical leadership of Dr. M. E. S. Loomis of D. Appleton Company, a substantial subset of LDDT was combined with the methodology of IDEF1 and published by the ICAM program in 1985. This technique was called IDEF1 extended or simply IDEF1/X.

The IDEF1X semantic data modeling technique was developed to meet the following requirements (IDEF1X 1993):

- *Support the development of conceptual schemas:* The IDEF1/X syntax supports the semantic constructs necessary in the development of a conceptual schema. A fully developed IDEF1/X model has the desired characteristics of being consistent, extensible, and transformable.
- *Be a coherent language:* IDEF1/X has a simple clean consistent structure with distinct semantic concepts. The syntax and semantics of IDEF1/X are relatively easy for users to grasp, yet powerful and robust.
- *Be teachable:* Semantic data modeling is a new concept for many IDEF1/X users. Therefore, the teachability of the language was an important consideration. The language is designed to be taught to and used by business professionals and system analysts as well as data administrators and database designers. Thus, it can serve as an effective communication tool across interdisciplinary teams.
- *Be well tested and proven:* IDEF1/X is based on years of experience with predecessor techniques and has been thoroughly tested both in Air Force development projects and in private industry.
- *Be automatable:* IDEF1/X diagrams can be generated by a variety of graphics packages. In addition, an active three-schema dictionary has been developed by the U.S. Air Force.

Use of this standard permits the construction of semantic data models, which may serve to support the management of data as a resource, the integration of information systems, and especially the building of computer databases. In an IDEF1X view, an “attribute” represents a type of characteristic or property associated with a set of real or abstract things (people, objects, places, events, ideas, combinations of things, etc.). An entity must have an attribute or combination of attributes whose values uniquely identify every instance of the entity. These attributes form the "primary-key" of the entity. In this way, the primary key or the combination of primary keys define a unique entity through which the attributes of its own and related entities can be reached. Figure 2.4 depicts the attribute and primary key syntax in IDEF1X.

In an IDEF1X view, connection relationships are used to represent associations between entities. A "connection relationship" (also referred as a "parent-child relationship") is an association or connection between entities in which each instance of one entity, referred to as the parent entity, is associated with zero, one, or more instances of the second entity, referred to as the child entity, and each instance of the child entity is associated with zero or one instance of the parent entity. A solid line depicts an identifying relationship between the parent and child entities as in Figure 2.5. If an identifying relationship exists, the child entity is always an identifier-dependent entity, represented by a rounded corner box, and the primary key attributes of the parent entity are also migrated primary key attributes of the child entity.

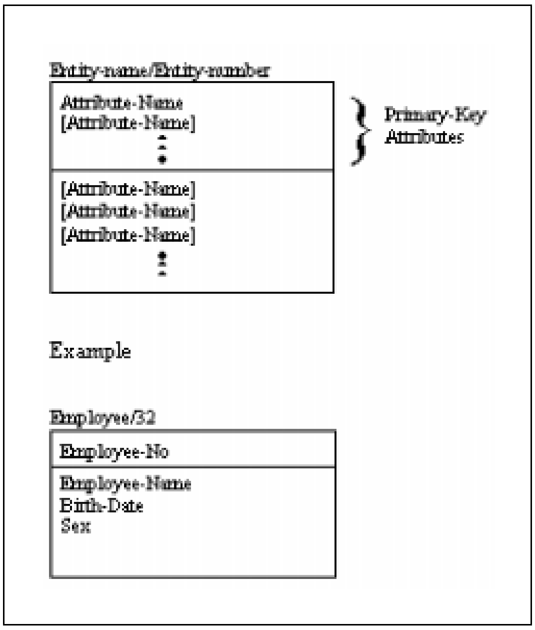


Figure 2.4 Attribute and primary key syntax in IDEF1X (IDEF1/X, 1993)

The IDEF1/X approach advises five phases for an analysis life-cycle:

- Phase Zero: Project initiation
- Phase One: Entity definition
- Phase Two: Relationship definition
- Phase Three: Key definitions
- Phase Four: Attribute definitions

IDEF0 and IDEF1X are powerful tools in modeling functionality, information flow and structure of an enterprise. If used together they form a full picture of working modules and their communication (Unver, 2000).

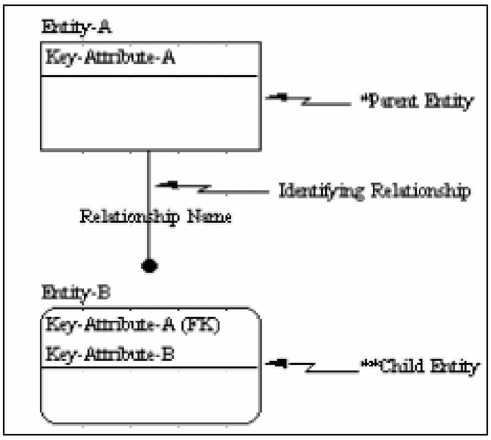


Figure 2.5 Identifying relationship syntax in IDEF1X (IDEF1X, 1993)

2.1.1.3 IDEF in the context of VE

IDEF modeling systems are industry-wide accepted standards today. Therefore several applications are developed in virtual enterprise domain by several researchers using this modeling technique.

IDEF0 function modeling has been used in OSMOS project to extract the generic view of the dynamic virtual enterprises. OSMOS, Open System for inter-enterprise information Management in dynamic virtual environments, is a European RTD project aimed at the development of an Internet based environment that supports share of information and task collaboration between non co-located teams in the context of extended enterprises, and especially in project based industries such as construction (Bourdeau et al, 2001).

The OSMOS project was driven by end-user requirements. A first essential workpackage was thus to produce a set of comprehensive requirements to be used as a basis for specifying the OSMOS generic solution. These requirements followed from an analysis of current business processes and information management practices within each of the three industrial partners involved in the project (Derbi, Granlund and JM). The stake was to understand current practices related to the establishment, maintenance and dissolution of a VE in the context of construction projects, as well as information flows and team work processes set up in this context, at all stages of the building lifecycle. A specific objective is to identify information inconsistencies and process inefficiencies currently taking place in collaborative work between non co-located teams on projects. In parallel, in order to understand the current partners' software and hardware infrastructures and requirements, a task is devoted to identifying common applications used within companies, and analyzing their internal data structures and information requirements.

Based on this analysis, generic models have been developed to describe basic processes taking place in a Construction Virtual Enterprise. The kind of interactions between actors and teams has been described and modeled, along with the nature and semantics of the information being produced and exchanged. These models are basic underlying elements for the specification of the OSMOS solution. The core modeling component is the Integrated Generic Process Model. It gives a generic view of the various activities required to operate a VE project. Figure 2.6 shows an extract of this generic VE process model, expressed with IDEF0 diagrams.

It describes the setting-up of a VE project. It should be noted that the lower levels of these diagrams has been translated into Use Cases descriptions.

GLOBEMEN (Global Engineering and Manufacturing in Enterprise Networks) project was initialized under the Intelligent Manufacturing Systems (IMS) program, in order to define and harmonize ICT support requirements in various one-of-a-kind industries operating in various cultural environments (Globemen, 2002). By combining the views and requirements of various industries the project aims to guide and encourage the industry and IT vendors to develop and adopt improved IT infrastructures. The focus is on inter-enterprise integration and collaboration in the three main facets of global manufacturing: sales and services, inter-enterprise delivery process management, and distributed engineering (using the…, globemen consortium).

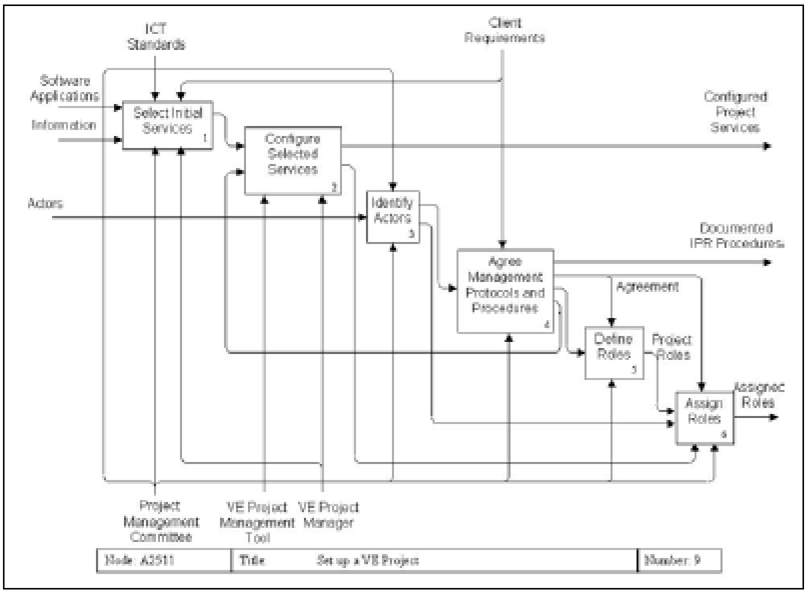


Figure 2.6 An extract of the integrated generic VE process model (Bourdeau, 2002)

In the light of core concepts presented above, the following main functions for product/service delivery through a virtual enterprise have been proposed:

- Operate enterprise
- Operate enterprise network
- Operate virtual enterprise
- Operate product collaboration

The main inputs, outputs, controls, and mechanisms for these core functions are represented in IDEF0 notation as shown in Figure 2.7. This high level IDEF0 diagram (Figure 2.7) was later validated by the industrial partners and specialized at several levels: domain level (e.g. distributed engineering), and industry level (e.g. construction). The captured data and their semantic relationships are modeled using the IDEF1/X modeling method.

IDEF0 has also been used on modeling of power plant construction by Globemen Fortum Engineering (Valikangas et al, 2000). Power plant construction typically involves virtual enterprises (VE), which are gathered from company-specific virtual networks (VN).

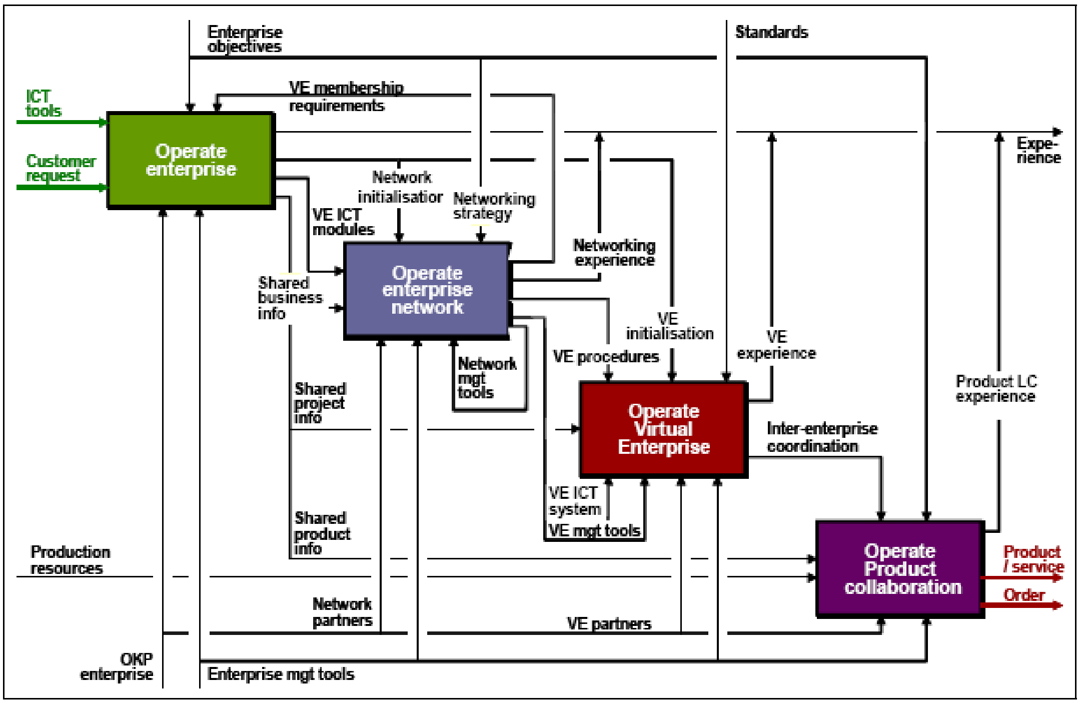


Figure 2.7 Core functions in IDEF0 format (Globemen, 2002)

The knowledge creation environment (KCE) has been developed for collecting and classifying tacit project-specific information and transforming it into explicit definitions. The role of KCE can also be seen as a way to artificially increase the amount and quality of data at the inception phase of the project, i.e. VE. This was done by combining the project-specific information with the hierarchically arranged experimental learning and product knowledge stored in KCE.

Managing power plant information is a complex activity covering the total lifecycle of the plant. Therefore, IDEF0 modeling technique has been used to outline the main phase in generating power plant specific information as shown in Figure 2.8.

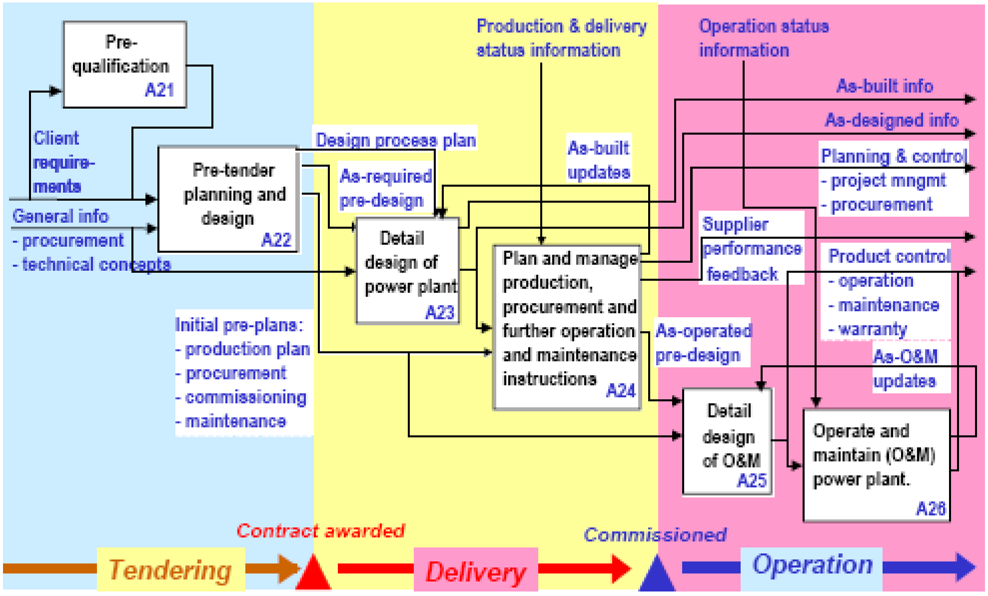


Figure 2.8 Functional model of power plant construction (Valikangas et al, 2000)

VITE (the fruit and vegetable business virtual enterprise) was another project which aimed at development of a virtual enterprise to have a reliable supply chain at lower costs in the fruit and vegetable sector (Noran, 2000). VITE combines participating farmers, wholesaler(s), retailers and trucking companies in an integrated system with the following features:

- Retailer enabled to remotely bid for products and check quality in real time;
- Farmers enabled to offer real-time info about products quality-quantity-availability;
- "Food Plant Type Company". (more a part of the wholesaler, seen as a separate entity for the sake of the model, in charge of the packaging, sorting, storing goods), independent or part of the wholesaler;
- Wholesalers allocating a part of their capabilities for the small/medium retailers;

also enabled to check availability, quality of the product from farmer;

- Small trucking companies able to respond to regular, peak, and on the spot product demand.

Functional model of the virtual enterprise is depicted in Figure 2.9 as in IDEF0 notation.

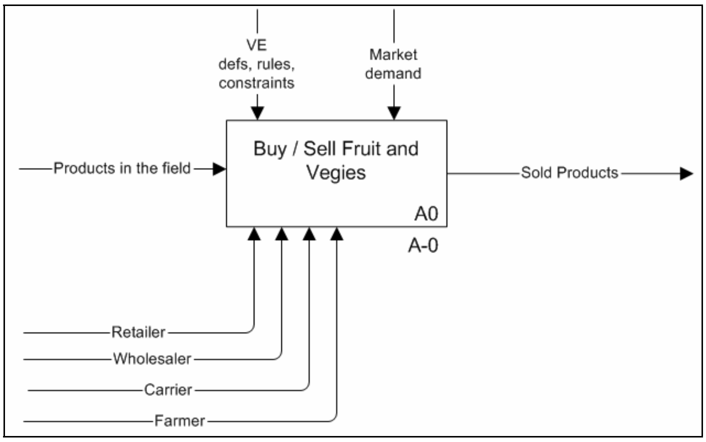


Figure 2.9 Functional model of VITE (Noran, 2000)

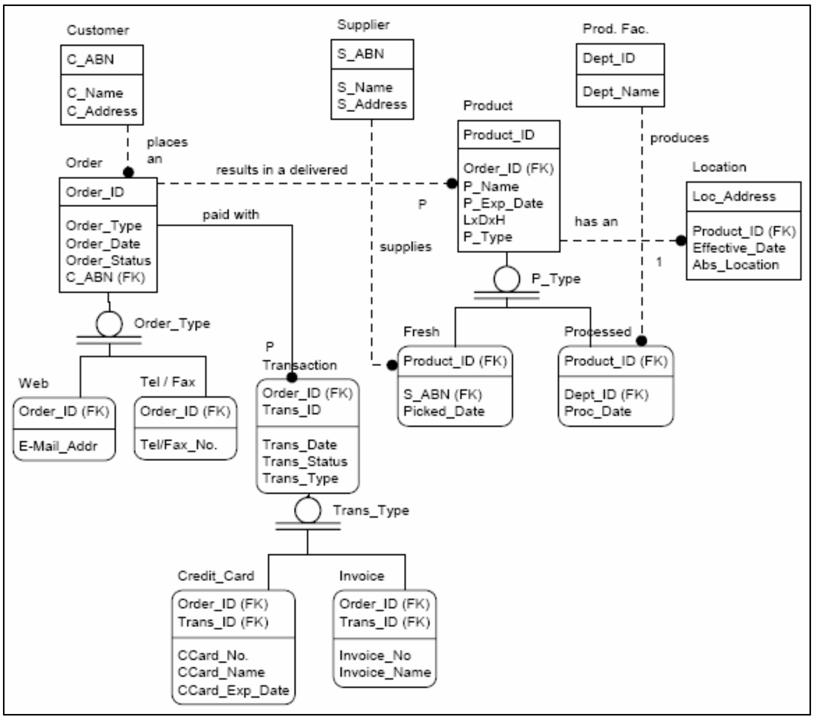


Figure 2.10 Information model of VITE (Noran, 2000)

The *controls* are the VE definitions, rules, constraints and market demand (fundamental control and reason for the VE to exist). The *mechanisms* are the farmer, carrier, retailer and wholesaler. VE *input* is the product in the field and the *output* is the sold product. The information model is developed based on IDEF1/X notation as shown in Figure 2.10.

2.1.2 Unified modeling language (UML)

Unified modeling language (UML) was developed by Grandy Booch, James Rumbaugh, and Ivar Jacobson in 1996 and later standardized by the Object Management Group in 1997. UML is one of the most powerful tools for modeling a dynamic system. The purpose of this section is to explain how UML fulfills its modeling promises through a set of notions/concepts and constructs.

UML is a very expressive language, addressing all the views needed to develop and then deploy systems ranging from enterprise information systems to distributed web-based applications and even to hard real time embedded systems. UML is used to visualize, construct, and document the artifacts of a software intensive system [Booch et al, 1999]. UML has a notation and a well-defined set of syntactic and semantic rules [Eriksson & Penker, 2000].

UML provides a common vocabulary of object oriented terms and a set of diagramming techniques that are rich enough to model any systems development project from analysis through implementation. It is nothing more than a notation and it does not dictate any specific approach to developing information systems.

Systems are modeled to:

- Provide structure for problem solving
- Experiment to explore multiple solutions
- Furnish abstractions
- Reduce time-to-market for business problem solutions

- Decrease development costs
- Manage the risk of mistakes

UML is a graphical language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML goals are:

- Define an easy-to-learn but semantically rich visual modeling language
- Unify the Booch, OMT, and Object modeling languages
- Include ideas from other modeling languages
- Incorporate industry best practices
- Address contemporary software development issues, which are scale, distribution, concurrency, executability, etc.
- Provide flexibility for applying different processes
- Enable model interchange and define repository interfaces

The strongest aspect of UML is its incorporation of views. UML takes into account five types of perspectives, or views, pertaining to any given piece of software. In the following sub-sections, these views, along with the UML graphical notation are introduced.

2.1.2.1 Functional view

The functional view represents how the user will view the software in terms of its functions. This view is constructed by **use-case diagrams**, which show the over-all functionality of software. It ignores how the software goes about performing its task and focuses on what is being performed. In Figure 2.11, the scenario where a student is enrolled in a university is considered.

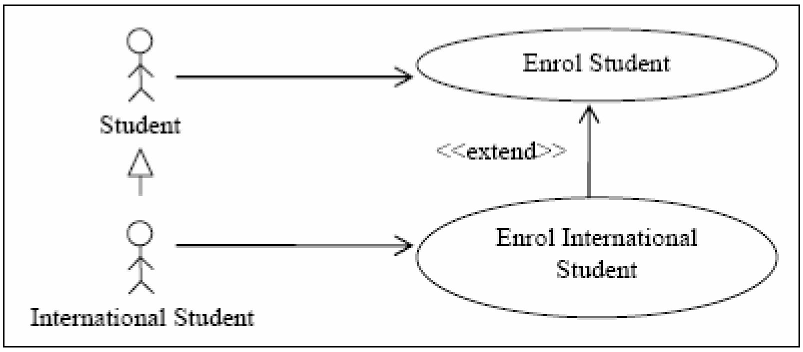


Figure 2.11 Use-case diagram for enrolling students (Rational, 1997)

The **Student** and the **International Student** are users and are represented diagrammatically as shown in Figure 2.11. The International Student 'is-a' Student, and this generalization association is represented by an arrow with a hollow triangle end, originating from the more specialized user, International Student in this case. **Enroll Student** and **Enroll International Student** are use-cases, which are performed for the users. The use-cases are represented diagrammatically as an oval shown in Figure 2.11. Enroll Student is extended into Enroll International Student and this extension relationship is represented by an arrow and a stereotype (see the next section) **<<extend>>**.

2.1.2.2 Structural view

A system is composed of classes, objects, and their associations. The structural view represents the system from these perspectives. This view does not display how the classes and objects actually behave, but shows their relationships. A structural view can be constructed with two types of diagrams, class and object diagrams.

Class diagrams: They describe the structure of a system. The structures are built from classes and relationships. The classes can represent and structure information, such as products, documents, and organizations. In Figure 2.12, a typical customer order is modeled using the class diagram. The classes are represented as rectangles with the names of the class written inside the rectangle.

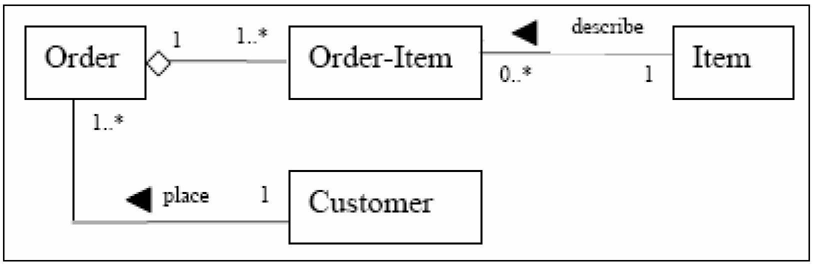


Figure 2.12 Class diagram for modeling an order (Rational, 1997)

In Figure 2.12, **Order**, **Order-Item**, **Item**, and **Customer** are all classes. The association name appears near the association line, and the multiplicities appear on each end. In Figure 2.12, **describe** and **place** are association names, and the numbers are the multiplicities. The customer places an order, which is shown by an arrow pointing towards order. The aggregate relationship is a specialization of association, where a whole is associated with its parts. The hollow diamond on the association line in Figure 2.12 denotes that an order aggregates many order-items. The multiplicity of the association can be represented as either a range (1..*), (0..*), etc., or a specific number (1). In the association between customer and order, the multiplicity icons represent the scenario where one customer places one or more orders. The semantics of the order-item and item classes with associations is that (i) all items and their definitions/descriptions are gathered with the Item class, (ii) not all items in the Item class are put on orders by customers.

Object diagrams: They express possible object combinations of a specific class diagram. They are typically used to exemplify a class diagram. Because an object diagram is an instance of a class diagram, multiplicity is not shown. The object names, which are underlined, are composed of their names, followed by a column and the class name.

In Figure 2.13, **AMX** object belongs to the class **Customer**. As UML is extendible, the object instantiated from the class is represented in this dissertation as shown in Figure 2.14.

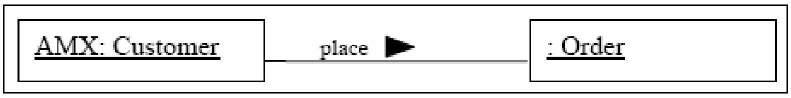


Figure 2.13 Object diagram for modeling a customer order (Rational, 1997)

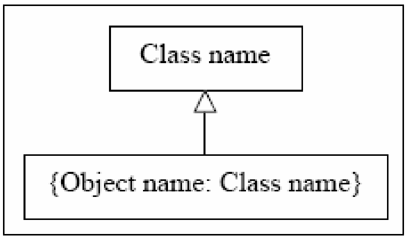


Figure 2.14 Object instantiated from class (Rational, 1997)

2.1.2.3 Behavioral view

This view represents the actual behavior of software. It contains diagrams representing the inner-workings of classes and their behaviors in respect to one another. The behavioral view is composed of four diagram types.

Sequence diagrams: They show one or several sequences of messages sent among a set of objects. The inter-class communication is broken down as a series of steps. Messages are shown as arrows that represent communication between objects. They follow similar association rules as mentioned in the earlier views. Lifelines are vertical dashed lines that indicate the object's presence over time. Activation boxes represent the time an object needs to complete a task and are represented by rectangular boxes along the lifelines. Objects can be terminated early using an arrow labeled **<<destroy>>** that points to an X. All these concepts are represented in Figure 2.15 with an example.

Collaboration diagrams: They describe a complete collaboration among a set of objects with their roles. Unlike sequence diagrams, collaboration diagrams do not have an explicit way to denote time; instead the messages are numbered in the order of execution. Sequence numbering can become nested, for example, nested messages under the first message are labeled 1.1, 1.2, 1.3, and so on. In Figure 2.16, the sequences of messages passed between different objects are represented.

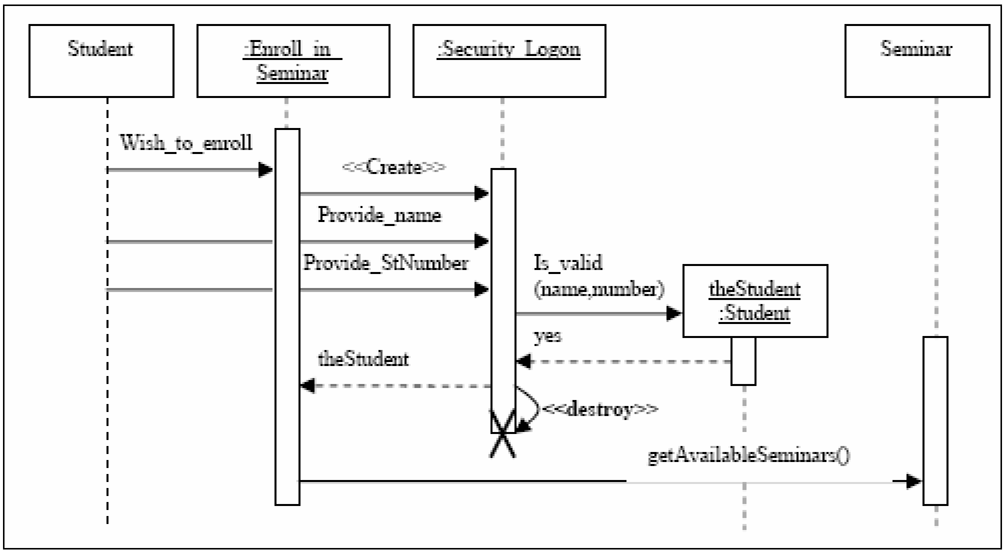


Figure 2.15 Sequence diagram for enrolling a student (Rational, 1997)

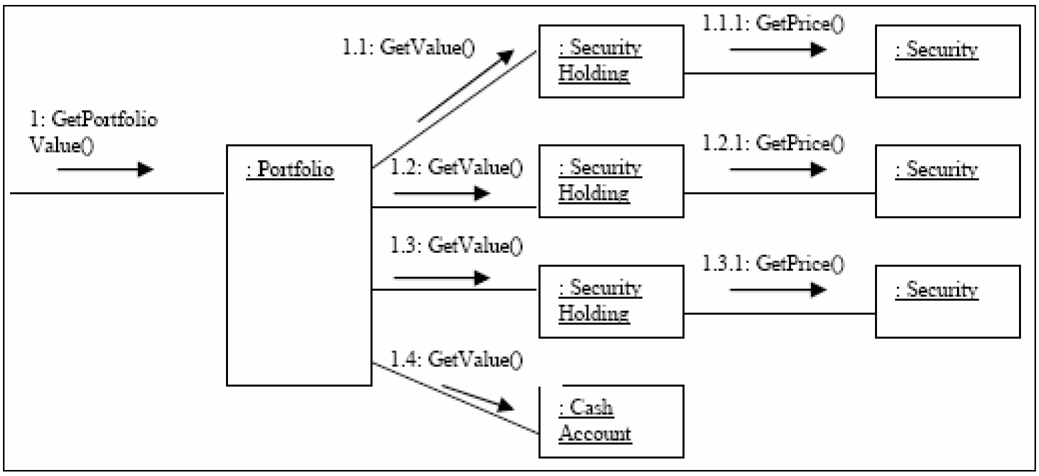


Figure 2.16 Collaboration diagram for the calculation of the value (Rational, 1997)

Statechart diagrams: They express object state under differing circumstances. External stimuli are the focus of statechart diagrams. States represent situations during the life of an object and can be illustrated using a rectangle with rounded corners. Changes in states are indicated with an arrow pointing from one state to another. The state transition is labeled with its cause. A filled circle followed by an arrow represents the object's initial state. An arrow pointing to a filled circle nested inside another circle represents the object's final state. The above-mentioned concepts are represented in Figure 2.17, which is a statechart diagram for invoices.

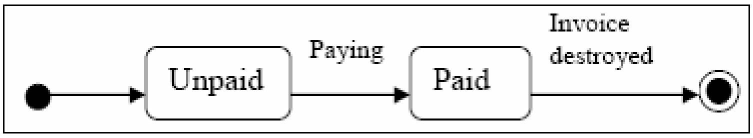


Figure 2.17 Statechart diagram for invoices (Rational, 1997)

Activity diagrams: they are much like statechart diagrams in that they express the response of classes with respect to execution, but activity diagrams focus on variables and states internal to a system rather than external stimuli. Activities are action states represented by a rectangle with rounded corners. The receiving activity is represented as a rectangle with a concave side and the sending activity as a rectangle with a convex side. Action flow arrows illustrate the relationships among activities. The initial and final states are represented as mentioned in the statechart diagrams section. A diamond represents a decision with alternate paths.

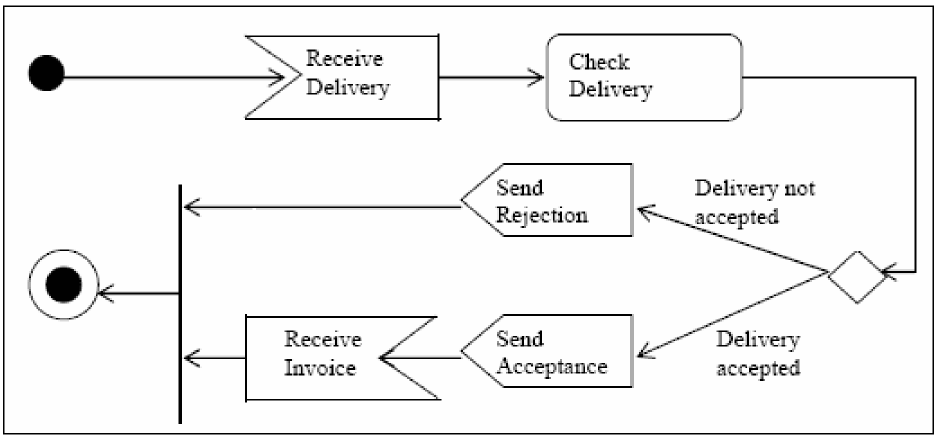


Figure 2.18 Activity diagram for receiving delivery (Rational, 1997)

In Figure 2.18, the two alternate decision paths, i.e., to **send rejection** or to **send acceptance**, are represented by a diamond. The outgoing alternates should be labeled with a condition or guard expression. A synchronization bar helps illustrate parallel transitions. In Figure 2.18, either the **send rejection** or the **receive invoice** activity could cause a final state in the activity.

2.1.2.4 Implementation view

This view demonstrates the organization and dependencies of the actual system and its pieces. This view has only one type of diagram.

Component diagrams: Show the organization and dependencies of a system's actual parts. The components are represented using rectangles with tabs and their relationships using dashed arrows as shown in Figure 2.19

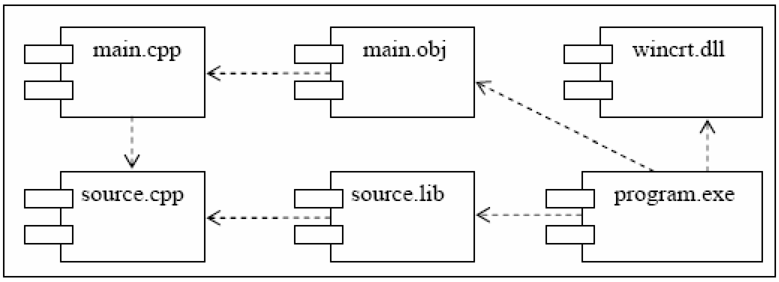


Figure 2.19 Component diagram between software components (Rational, 1997)

2.1.2.5 Environment view

This view represents the environment in which the software will exist when finished. This view has one diagram type.

Deployment diagrams: Show all required resources for the software. This is a special case of class diagram used to describe the hardware within a software system. The nodes are represented by cubes and their associations by lines, as shown in Figure 2.20.

2.1.2.6 UML in the context of VE

UML modeling methodology has been widely used in distributed engineering domain for modeling a set of business processes of the virtual enterprise systems.

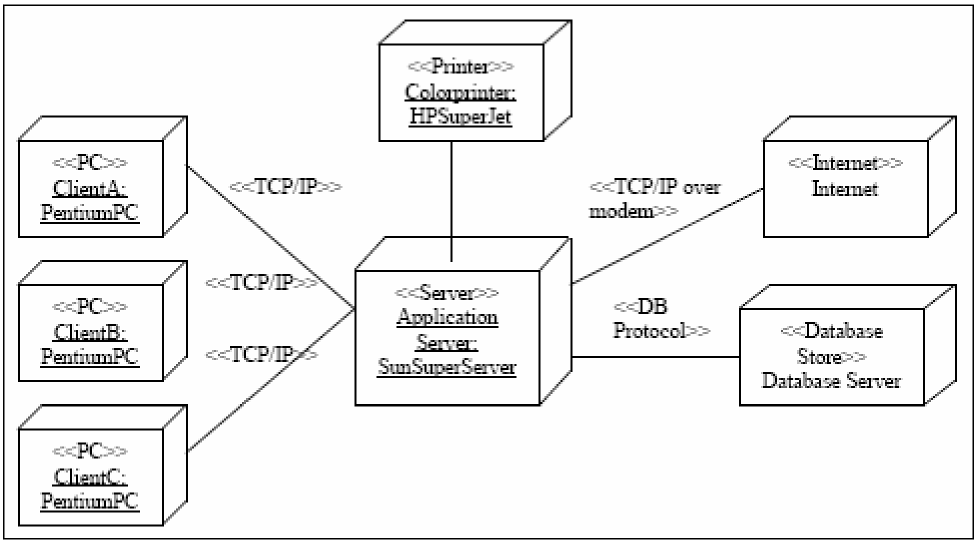


Figure 2.20 Deployment diagram of physical hardware (Rational, 1997)

In VITE project, part of the business processes has been modeled using UML (Noran, 2000). VITE is a virtual enterprise aimed at selling fruit and vegetables. It combines participating farmers, wholesaler(s), retailers and trucking companies in an integrated system. The conceptual model defining the business key concepts and taking the form of class diagram is given in Figure 2.21. The business plan (delivered by a business development process) contains:

- Product strategy – deals with the farm products, product packaging / storage, it is derived from market analysis and results in definitions of product / product sets;
- Marketing plan – based on market analysis may influence the business plan;
- Business ideas are important business assets
- Internet strategy – is a main component of the (VE) integration process

A sequence diagram is also used in order to analyze a typical scenario (e.g. stock purchasing). Interactions are triggered by a business event and end with a result. In the scenario shown in Figure 2.22, the purchase department has to interact with the other organizational units in order to determine what to purchase, when to make the purchase, and how much to purchase based on trends and forecasts.

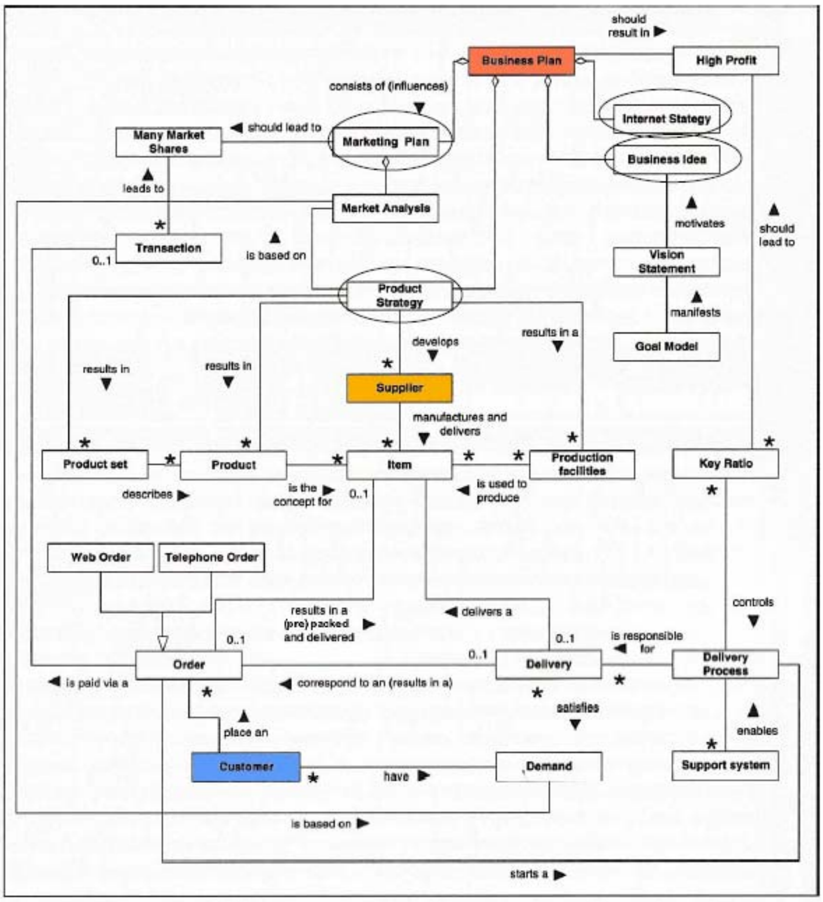


Figure 2.21 Conceptual model (Noran, 2000)

Besides the presented diagrams, activity diagrams, use cases and state diagrams are also used to establish the full model of the system. Hongmei et al. have used multiple UML diagrams to specify and design the static model (by UML class diagrams and packages) and the dynamical model (by UML sequence diagrams, collaboration diagrams, activity diagrams and statechart diagrams) of the business process in the VE (Hongmei et al, 2000).

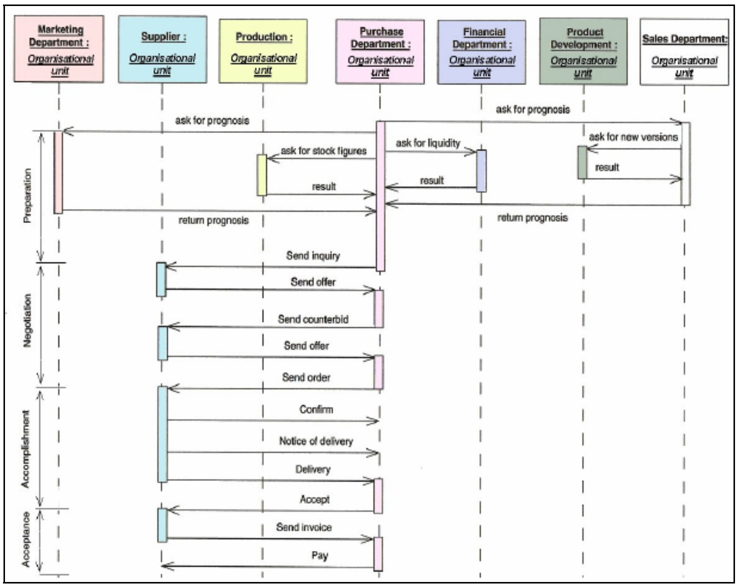


Figure 2.22 Sequence diagram (Noran, 2000)

UML class diagrams are used to describe business objects (including properties and operations) and their associations. In Figure 2.23, the top-level class architecture of our demonstrative VE is described by the UML class diagram and operations of classes are labeled as P01~P23.

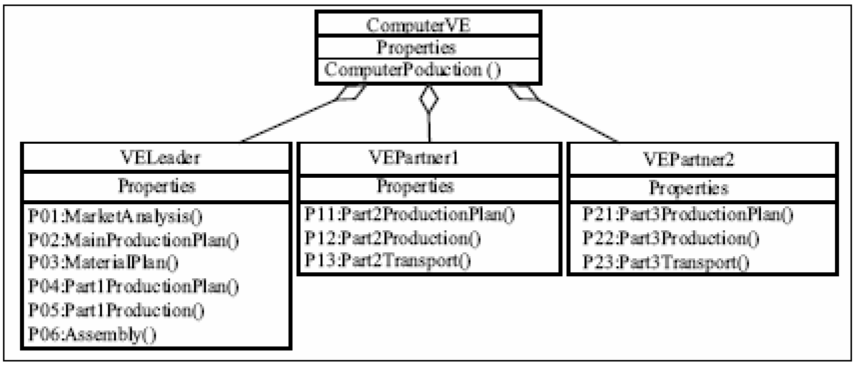


Figure 2.23 UML class diagram for business objects (Hongmei et al, 2000)

UML packages are used to describe organization patterns of the VE. UML Sequence diagrams and collaboration diagrams are used to describe interactions among business objects. Based on message mechanism, state transitions of business objects are triggered by message transfers, which can be viewed as events (called as message events later). Business objects also send messages to themselves, therefore changing states of their own. In Figure 2.24, message transfers inside/among business objects are described by the UML sequence diagram and message events are labeled as t1~t11.

UML activity diagrams are used to describe business activities and their transitions from global view of the whole business process. As business activities correspond to operations in class diagrams, such decomposition actually can be achieved after the establishment of class hierarchy. In Figure 2.25, relations among business activities are described by the UML activity diagram.

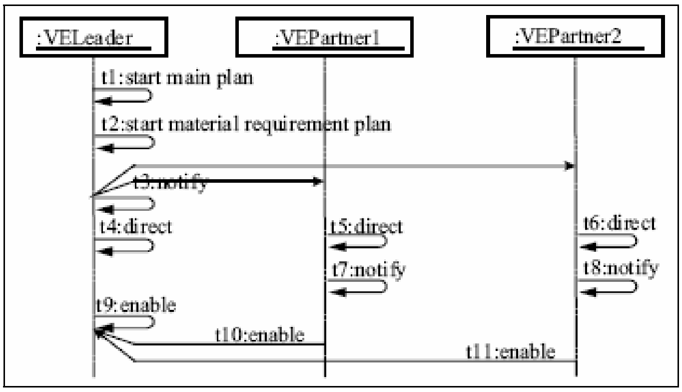


Figure 2.24 UML sequence diagram for message transfers (Hongmei et al, 2000)

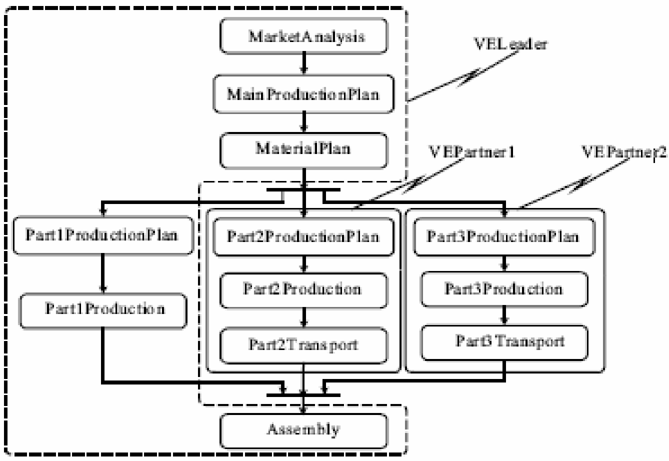


Figure 2.25 UML activity diagram for the business process (Hongmei et al, 2000)

UML statechart diagrams are used to describe states and transitions inside a business object. States here also correspond to operations in class diagrams. All the states in statechart diagrams of all the objects constitute the set of business activities. Therefore, statechart diagrams describe interrelations of business activities from the object view.

2.1.3 Entity-relationship modeling

E-R modeling (Chen, 1976) is a technique of conceptual modeling which produces a high-level formal model of the organization. This model is then used as a framework for current and future development of information systems. The model is independent of process and in general provides an unambiguous way of presenting ideas.

The model has a number of components. An entity is a thing of interest to the organization and something about which we want to hold information. An entity instance might be some physical object, for example, a person, or it might be a conceptual object, for example, a job, about which we need to store information. An entity is represented by a box with a name as shown in Figure 2.26.

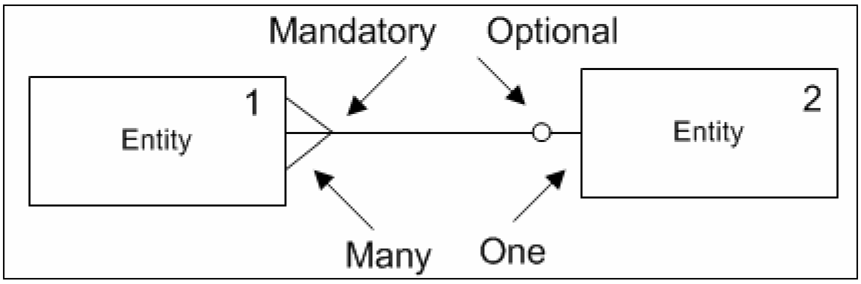


Figure 2.26 Entity-Relationship notation (Chen, 1976)

Relationships associate or relate instances from one entity type with some of the instances of another entity type. Relations occur because of either an association or a structure. Each relationship has a name, degree or cardinality (how many) and a membership class (optional or mandatory).

A relationship is represented by a line that joins entity boxes together. In a recursive relationship the line joins an entity box to itself. An important property of a relationship is its degree (Figure 2.26). There are three possible kinds of relationship degree:

- One-to-one (1:1) An example to this is that a worker may only work at one department in a factory.
- One-to-many (1:M) An example to this kind of relation is that a process plan will have more than one operation.
- Many-to-many (M:M) An example of this type relationship could correspond to a business rule that a CNC may be alternative to many operations and an operation may have many alternative CNCs.

The membership class of a relationship denotes whether it is optional or mandatory for every occurrence of the related entities to participate in an occurrence of the relationship. Where the relationship end is optional, a small circle is placed on the line at the appropriate end of relationship.

2.1.4 Object oriented analysis and design

There is a wide range of object-oriented methodologies, each with its own view of development process. Some focus on design and implementation, some on analysis, some concentrate on a particular modeling technique, while others are specifically geared towards a particular programming language. However, there is one feature that all these methodologies have in common: they organize the development of systems around objects. New methodologies are constantly being developed and old ones are continually being updated, so it is not possible to cover all object-oriented methodologies at one time.

2.1.4.1 Objects

The basic building block of object-oriented software is the object. Software objects are derived from and model the real-world object in the application domain. That is, whatever type of software system you are developing, whatever problem you are trying to solve, it will feature certain entities, objects or things.

In an early book on object-oriented analysis and design, Coad and Yourdan (1990) define object as follows:

> "*Object. An abstraction of something in a problem domain, reflecting the capabilities of the system to keep information about it, interact with it, or both.*"

Rumbaugh et al. take a slightly different slant on defining an object (1999):

> "*We define an object as a concept, abstraction, or thing with crisp boundaries and meaning of the problem in hand. Objects serve two purposes: They promote understanding of the real world and provide a practical basis for computer implementation.*"

Objects in O-O systems can represent physical things (such as products, parts, customers, and partners) or organizational things (such as department or companies) or even computer implementation features such as files or linked lists. A brief terminology of Object-Oriented systems is given in Table 2.1 and they will be described in detail in the following sections.

Table 2.1 Object-oriented terminology (Britton and Doake, 2005)

Term	Definition
Object	Software unit packing together data and methods to manipulate that data
Class	Template or factory for creating objects
Attribute	Data item defined as part of a class or object
Operation	Procedure or function defined as part of a class or object; using this term refers to the procedure's public interface with the rest of the software
Method	Procedure or function defined as part of a class or object; using this term refers to the procedure's implementation
Message	Request sent to an object to execute one of its methods
Encapsulation	Packing data and operations into an object
Data hiding	Making the internal details of an object inaccessible to other objects
Inheritance	Mechanism for defining a new class in terms of an existing class
Polymorphism	The ability to hide different implementations behind a common interface

2.1.4.2 Class

Classes are templates that have methods and attributes and type information, but no actual values (Figure 2.27). Objects are generated by the classes and actually contain values. Some classes are very abstract and are simply used to provide structure for other classes. Most classes, however, are templates that are used to generate the objects that actually do the work when application runs. A class can be thought as an object factory and they are generally represented as rectangles with three sections: one for the name of the class, one for the attributes of the class and one for the operations of the class (Harmon and Sawyer, 2004).

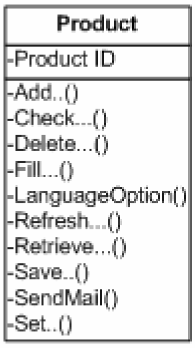


Figure 2.27 A class with attributes and operations (Harmon & Sawyer, 2004)

2.1.4.3 Message

The overall functionality in an object-oriented system is achieved by object interacting with each other. They do this by sending messages to each other requesting services. The initiating object is referred to as the client or sender, the receiving objects as the server or receiver. When a server object receives a message it executes the appropriate operation.

Message passing is important because it represents two definitive aspects of the O-O approach to software design. In procedural code, data are passed between functions, for example a sine function will be sent an angle; will calculate its sine and send back the answer. In an O-O program the sine function is an operation defined on the class Angle and therefore all objects of the Angle class will know about this operation and be able to calculate their sine on demand. The other aspect is the emphasis placed on building objects that are independent and self-sufficient. Objects once activated by a message should take charge of executing their responsibilities. For example, a Product object having CheckProgress function activated by a message should send a message to its related components (parts or tasks) and check the progress achieved so far in terms of % work completed.

2.1.4.4 Encapsulation

Packaging related data and operations together is called encapsulation. The encapsulation mechanism provides three desirable software qualities:

- *Modularity:* The encapsulation of data and operations into a single software construct- the object- provides useful building block of software systems. This software construct provides software modules ideal for producing reusable units of code. The objects are designed to be as independent as possible, so that they should not have many dependencies on other parts of the code and should be easily transformed into candidates for a software library.
- *Data abstraction:* An object's name, class and the names of its operations form its public interface. This provides the abstraction – all the object's clients need to know about is the information provided in the public interface while the internal representation of the object's data and operations are irrelevant and invisible to the client.
- *Data hiding:* The data encapsulated in an object can only be accessed by using that object's operations, operations designed to ensure that the data is handled properly. We can think of the data as being surrounded by a protective outer ring of operations. Client modules cannot access that data directly, they can only send a message to the object requesting that it execute on of its methods. This method in turn will access the data in the proper way. Encapsulation ensures that the internal representation of the data and operations is concealed. The external interface of an object does not tell the world details of how data is stored or what algorithm is used by a method. This is known as data hiding.

2.1.4.5 Inheritance

Inheritance is a mechanism, which allows new classes to be defined in terms of existing classes; a new class can be defined as a specialization of one that has already been written. The specialized class automatically includes or inherits the features (operations and attributes) of the class it has been created from. An inheritance relationship is indicated by a line between classes with a semi-circle pointing towards the parent class. Conventionally, inherited features are not shown in the specialized classes, but features, which are introduced in the specialization, are shown (Figure 2.28).

2.1.4.6 Polymorphism

The term polymorphism means the ability to define program entities like operations that take more than one form. Polymorphism linked with an inheritance hierarchy allows a single message to be interpreted differently by different objects. Which method is executed will depend on which object receives the message.

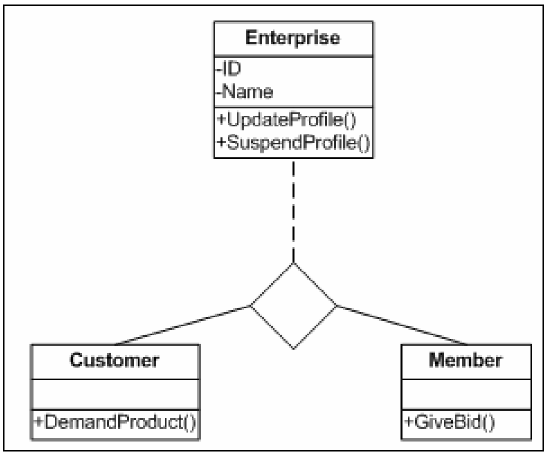


Figure 2.28 Inheritance mechanism

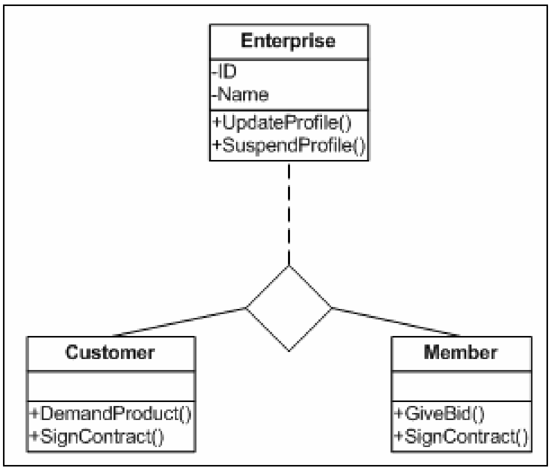


Figure 2.29 Polymorphism implemented in class hierarchy

For instance, two different Enterprise subtypes, Customer and Member will respond to SignContract operation by signing product contract for Customer and task contract for Member (Figure 2.29).

2.2 Advances in Software Technologies

2.2.1 Information systems

An information system provides procedures to record information and make it available, concerning part of a organization, to assist organization-related activities (Flynn, 2003). The aim of the information system is to provide a means for processing information to improve the efficiency and effectiveness of the organization. Information systems can be classified into three types:

- Informal information systems: these are the evolving complex patterns of behavior within organizations, which are never formulated, but which need to be understood by all those involved within the organization. These informal information systems can be vital to the organization's effectiveness.
- Manual information systems: these are the formalized procedures, which are not computer based, for producing information within an organization. These were the information systems before 1960s. It was about that time computers began to be used by organizations.
- Computer-based information systems: these are automated procedures for producing information using information technology, that is, computer hardware, software, and telecommunications, to help this process.

The computer based information systems can be sub divided into many groups:

- Data and transaction processing systems: these involve the use of computers to process the routine transactions of the organization. Such systems are used, for example to process accounts, purchasing, product and personnel data.
- Office automation systems: these use computers to automate general office tasks.

They usually consist of software for typing, processing, storing and retrieving documents electronically; software for managing databases, such as addresses and contacts; and software for electronic mailing.

- Management information systems: these use data from the data processing system to produce summary information about business performance. A management information system for a chain of supermarkets, for example, might produce an aggregate data relating to its entire stores.
- Decision support systems: these are extended management information systems into which hypothetical information or projections of information can be input to enable "what if" analysis and hypothesis testing.
- Executive information systems: designed for top management, these tend to be highly graphical in nature, permitting easy access to a wide range of information.
- Collaborative information systems: connecting independent companies via these systems creates process integration, improves forecasting and product planning and provides real-time access to order and shipment status – ultimately reducing manufacturing, distribution and sales costs. This type of information systems facilitate direct trade in business critical goods, with community participants creating symbiotic relationships that extend well beyond the buy/sell process.

Information systems play a crucial role in the success of organizations. They derive benefits from the information they provide, such as:

1. Efficient operations: where efficiency is considered to be the maximization of throughput with respect to the unit of resource input, that is, the organization obtains maximum benefit with the least waste from the various resources it allocates to tasks.
2. Effective management: where effectiveness is the ability to produce the intended output in a satisfactory manner. It is one measure, for example, of how well the products and services of an organization meet customer needs.
3. Competitive advantage: where, having already used information technology for producing information to the operational and management activities of the business efficient and effective, the organization uses information in a new and

innovative way to improve business performance, cut costs and to have an advantage in comparison to competitors.

2.2.1.1 Era of internet

Powerful desktop PCs and powerful servers, and client/server software started ruling the world since the end of 20^{th} century. People start shopping on Internet, deciding what to wear or what to cook with Internet. Tremendous development in distributed systems in parallel leads to distribution and processing of knowledge on Internet (Drucker, 1999).

The emergence of Internet let the industry into the second generation, represented by the web-presence (including using internal web-based productivity tools, and external web-based catalogs) and e-commerce exchanges (B2C or B2B). While Internet based commerce greatly improves productivities, and provides certain level of efficiency, it is fundamentally about competition - putting buyers against sellers to get the best price for indirect materials or excess inventories. These e-commerce exchanges aggregate catalogs, facilitate auctions, and bring large number of buyers and sellers together via the Internet (Dell, 1999).

2.2.1.2 Internet architecture

The Internet architecture evolved out of experiences with an earlier packet-switched network called the ARPANET. Both the Internet and the ARPANET were funded by the Advanced Research Project Agency (ARPA), one of the R&D funding agencies of the U.S. Department of Defense. The Internet and the ARPANET were around before the OSI architecture, and the experience gained from building them was a major influence on the OSI reference model (Peterson and Davie, 2000).

While the seven-layer OSI model can be applied to the Internet, a four-layer model is used instead. At the lowest there are a wide variety of network protocols. In practice, these protocols are implemented by a combination of hardware and software, i.e. an

Ethernet or Fiber Distributed Data Interface protocols. The second layer consists of a single protocol (IP). This is the protocol that supports the interconnection of multiple networking technologies into a single, logical network. The third layer contains two main protocols, the Transmisssion Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP and UDP provide alternative logical channels to application programs: TCP provides a reliable stream channel and UDP provides an unreliable datagram delivery way. Running above the transport layer are a range of application protocols, such as FTP (File Transfer Protocol), TFTP (Trivial File Transfer Protocol), Telnet and SMTP (Simple Mail Transfer Protocol), HTTP (Hyper Text Transfer Protocol) that enable the interoperation of popular applications.

2.2.2 Database management systems

The objective of a database management system (DBMS) is to provide mechanisms for managing integrated, shared data repository and to make the data easily accessible to multiple users while maintaining data integrity, availability, and security. Repository data model entities in the real world are customers, products, parts, etc. The way in which real-world entities are modeled depends on how the DBMS structures the data, which represent it. A data model is a collection of well-defined concepts to reason about data and relationships. A schema defines the static properties such as entities, relationships, and constraints of a database and operations define the dynamic properties.

The idea of storing data records in files as part of a simple file system and of linking data records to represent more complex data structures evolved in the late 1950s. In 1959, the Conference on Data Systems Languages (CODASYL) established two committees, one, which developed the COBOL business programming language, the other to investigate database concepts (named CODASYL Database Task Group (DBTG)). In 1969, the DBTG published its specification of the network data model and the data manipulation language. In the late 1960s, Ted Codd of IBM developed a programming language, which became the basis of much of relational database theory and eventually three classes of DBMS evolved (Simon 1996):

- Hierarchical
- Network
- Relational

Whatever the data model employed, the fundamental goals of a DBMS are as follows (Simon 1996):

- Data independence: Changes to the underlying physical storage of data and access mechanisms should not affect any programs that access the data. Conversely, an application program can be changed without affecting any other programs.
- Normalized data: A database should not contain duplicate or redundant data.
- Data integration: No inconsistencies or inaccuracies should arise as a result of changes of data.
- The business view of data should be separated from the physical representation of data on some storage device. A logical data model represents data as the user wishes to see it whereas a physical data model represents the structure of the data as it is stored on a storage device.
- Data access should be subject to security restrictions and concurrency of access controlled while maximizing the potential for data sharing.

2.2.2.1 Hierarchical data model

Data is organized hierarchically as a collection of data records connected to one another according to parent-child (one-to-one) relationship forming a tree-like structure. A hierarchical database is therefore an ordered set of trees with the use of pointers between records in order to traverse a tree starting at the root/parent node (Figure 2.30). In addition to defining the data, the navigational paths to data need to be defined as pointers and embedded in the database. Access to data is directly done by the user using data manipulation language commands, which guide the DBMS in locating relevant data. The use of pointers improves performance but reduces access flexibility and the user has to be very familiar with the structure of the database in order to access data. Another drawback is the need to duplicate data in order to model some complex relationships. This may result in unnecessary waste of storage space and potential data inconsistency when updating takes place.

2.2.2.2 Network data model

The network data model is the extension of hierarchical data model in that each data record type can be associated with one or more different record types using a system of pointers which form an arbitrary graph rather than a tree-like relationship structure. Parent-to-child relationships between record types are modeled as separate structure called sets (Figure 2.31). A set identifies links between one parent and one or more record types. A child record type can have more than one parent. In this way more complex relationships can be modeled more flexibly involving less data redundancy than hierarchical data model. Access paths are also pre-defined using pointers and are traversed using data manipulation language commands. Data access can be very rapid, but the drawback is bad, it is not easy to change the database structure without affecting the application programs that interact with it (Hernandez, 2001).

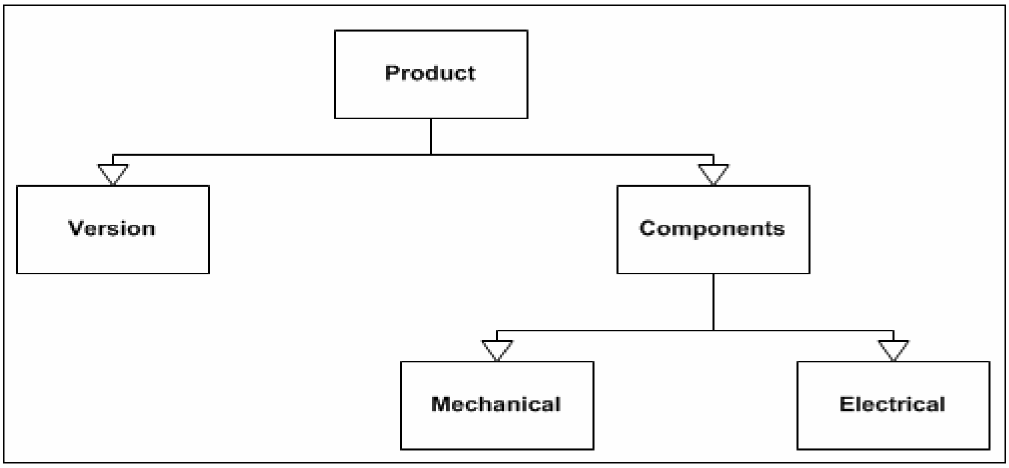


Figure 2.30 Hierarchical data model

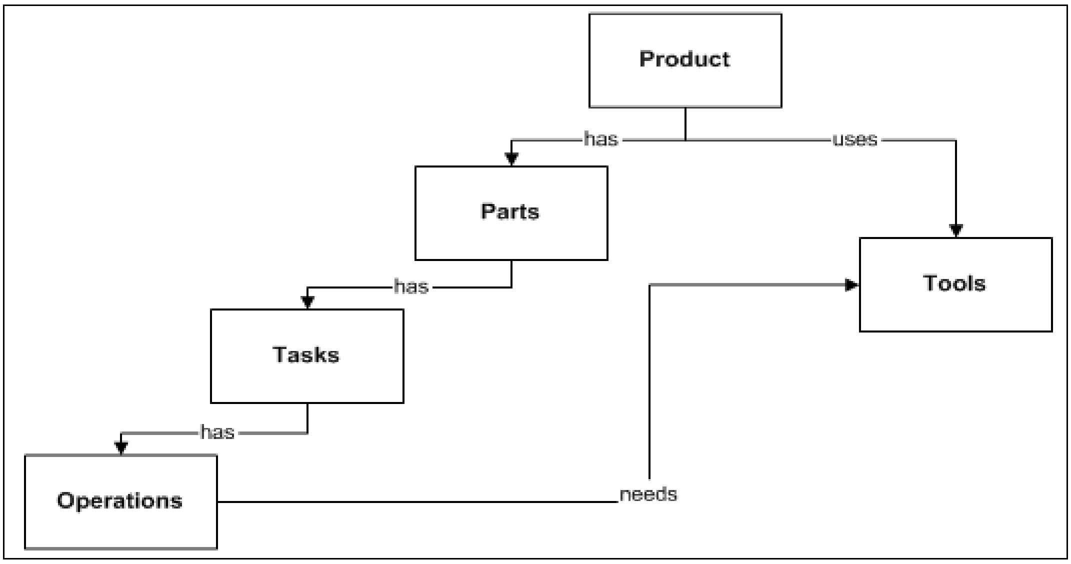


Figure 2.31 Network data model

2.2.2.3 Relational data model

In the relational data model all data are represented by only one data structure: two-dimensional tables (Figure 2.32). A table consists of a row of column headings and zero or more rows of data values. The intersection of a row and column identifies a single value. A primary key is specified for the table and is used to identify each row in the table uniquely. For primary key columns, duplicate values are forbidden and no part of the primary key may be a null value. Relationships between tables are established by normalizing initial table structures to identify primary keys and remove data redundancy. A foreign key is an attribute common to two or more tables and which is a primary in an associated table (Hernandez, 2001).

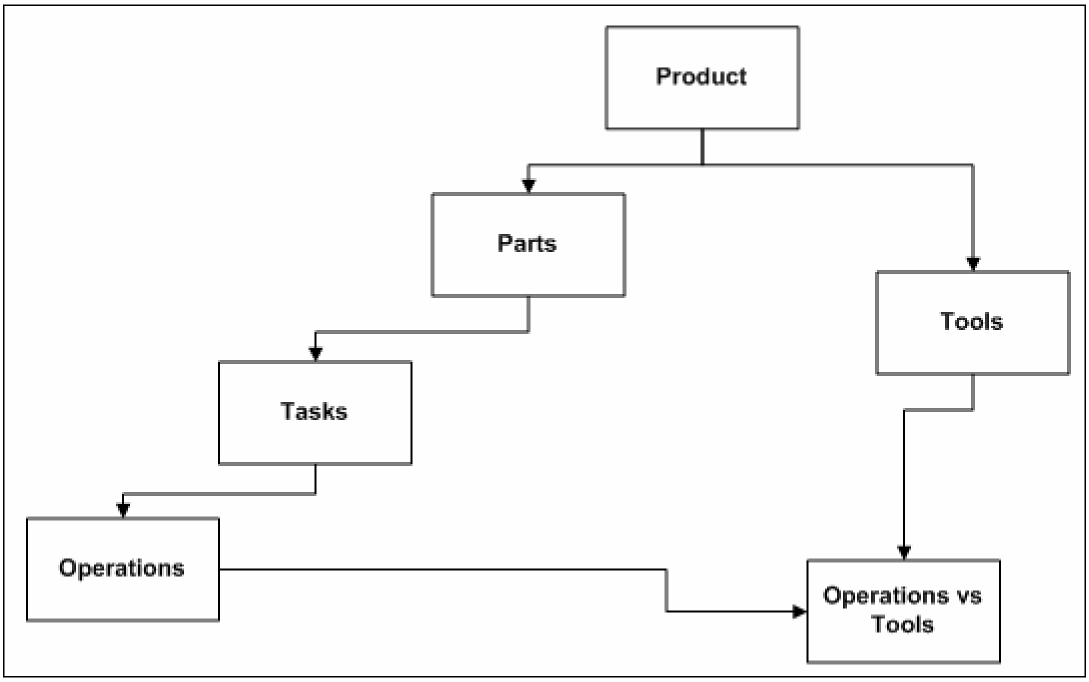


Figure 2.32 Relational data model

2.2.2.4 Structure query language

There are three major aspects of relational DBMSs: tables, representation of relationships through primary and foreign keys and use of relational data manipulation language. Codd’s initial work in 1970 resulted in a definition of relation data language called SQUARE and later SEQUEL. In 1982, IBM produced its first commercial relational DBMS SQL/DS which used a relational data language called structured query language or in short SQL. SQL has become the de facto standard language for accessing both relational and non-relational DBMSs.

In 1982, ANSI and ISO began to work on standardizing SQL. ANSI SQL was produced in 1986 known as SQL-86. ANSI-89 standard followed and then later SQL-92. ANSI defined three categories of SQL commands (Simon 1996):

- *Data definition language (DDL):* These are operations for creating, altering and deleting tables. DDL operations are normally restricted to a database administrator or application developer to ensure data integrity and security. The SQL CREAT command performs the main DLL operation to set up a data structure. Other commands are ALTER and DROP for structures.

- *Data control language (DCL):* These provide data security operations. Operations include the definition of VIEWS to restrict an application's window on a particular table by specifying which columns and rows can be seen by the application. Access to tables can be restricted further by use of GRANT and REVOKE facilities, which impose, access limitations on special users. Protection from partial failure of SQL operations is implemented via COMMIT and ROLLBACK processing which allows applications to ignore the effects of other applications that may be accessing shared data concurrently and provides a recovery mechanism after application and system failures.
- *Data manipulation language (DML):* These are operations for querying, insertion, deleting, and modifying data values on the database. Data manipulation is by means of four fundamental operations each of which returns a virtual table called a result set, holding the results of the operation, projection, join and concatenation. A selection operation returns a result set which, based on a specified selection criteria, returns a subset of table rows. A projection operation selects a subset of the columns defined for a table. A join operation joins two or more tables together based on specified row selection criteria. Concatenation operations combine two tables according to criteria based on the set operators: union, difference, intersection and divide.

2.2.3 Microsoft .NET framework

2.2.3.1 What is Microsoft .NET?

Microsoft began its Internet development efforts in 1995. Prior to this, Microsoft's focus had, for several years, been on moving desktop and server operating systems to 32-bit, GUI-based technologies, but once Microsoft realized the importance of the Internet, it made a dramatic shift. The company became focused on integrating its Windows platform with the Internet and it succeeded in making Windows a serious platform for the development of Internet applications.

However, it had been necessary for Microsoft to make some compromises in order to quickly produce Internet-based tools and technologies. The most glaring example was Active Server Pages (ASP). While ASP was simple in concept and easily accessible to new developers, it did not encourage structured or object-oriented development. Creating user interfaces with interpreted script and limited visual elements was a real step back from the form-based user interfaces of Visual Basic (VB). Many applications were written with a vast amount of interpreted script, which lead to problems of debugging and maintenance.

Visual Basic (and other languages) has continued to be used in Internet applications on Microsoft platforms, but mostly to create components that were accessed in ASP. Before Microsoft .NET, Microsoft tools were lacking in their integration and ease-of-use for web development. The few attempts that were made to place a web interface on traditional languages, such as WebClasses in VB, were compromises that never gained widespread acceptance. The result was that developing a large Internet application required the use of a large number of loosely integrated tools and technologies.

Microsoft .NET is the first software development platform to be designed from the ground up with the Internet in mind – although .NET is not exclusively for Internet development; rather it provides a consistent programming model that can be used for many types of applications. However, when an application needs Internet capabilities, access to those capabilities is almost transparent, unlike tools currently used for Internet-enabled applications.

To understand what the importance of .NET is, it's helpful to understand how current tools such as ASP limit us in a development model based on the Internet. In this section, it will be looked at what's wrong with the DNA architectural model, and then examine how .NET corrects the drawbacks in these technologies. Although the drawbacks of current tools in the context of the Microsoft platform will be discussed, almost all apply in some form to all the platforms that are currently available for Internet development. Moreover, many of these other platforms have unique drawbacks of their own.

2.2.3.2 The DNA programming model

In the late 1990s, Microsoft attempted to bring some order to Internet development with the concept of Windows DNA applications (Microsoft, 1999). DNA consists of a standard three-tier development based on COM, with ASP (as well as Win32 clients) in the presentation layer, business objects in a middle layer, and a relational data store and engine in the bottom layer. Figure 2.33 shows a generic Windows DNA application.

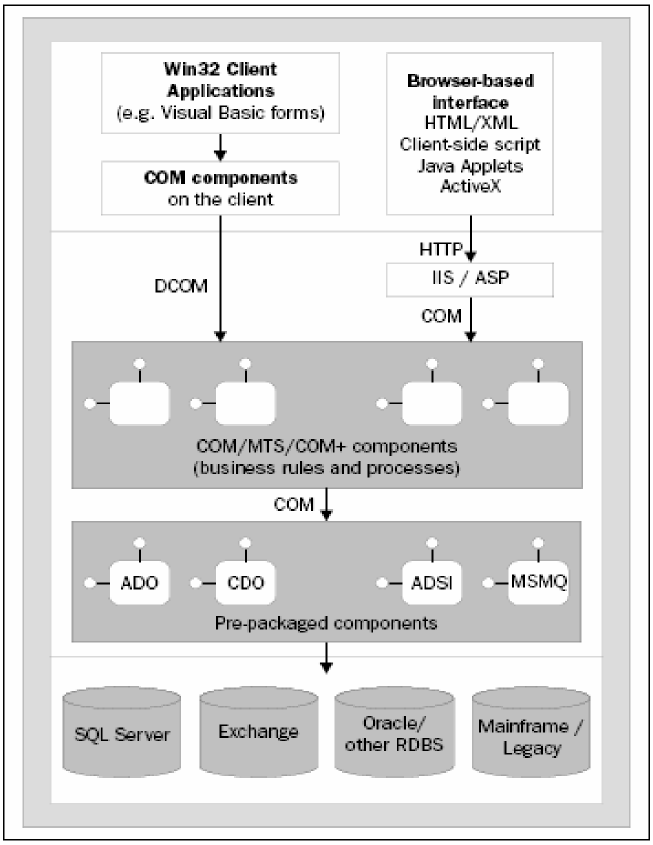


Figure 2.33 Generic Windows DNA application

DNA architecture maps out the framework for building scaleable, three-tier distributed applications that can run over any network, including the web. The term “three-tiered” defines the computers, on which the application/service is running, which are:

- **Client tier**: a local computer on which either a Web browser displays a Web page that can display and manipulate data from a remote data source, or (in non–Web-based applications) a stand-alone compiled front-end application.
- **Middle tier**: a server computer that hosts components that encapsulate an organization's business rules. Middle-tier components can be either Active Server Page scripts executed on Internet Information Server, or (in non–Web-based applications) compiled executables.
- **Data source tier**: a computer hosting a database management system (DBMS), such as a SQL Server database. (In a two-tier application, the middle tier and data source tier are combined.)

2.2.3.3 Limitations of DNA internet development

There are a few additional areas in which previous Microsoft tools and technologies fell short of the ideal for Internet application development.

2.2.3.3.1 Different programming models

With DNA-based software development, creating software that is accessed by a user locally is done very differently from development for the Internet. The starkest example of this is the use of VB forms for client-server user interfaces versus the use of ASP for Internet user interfaces. Even though both situations involve designing and implementing GUI-based user interfaces, the tools and programming techniques used are quite different.

Having very different programming models for these similar types of development causes several problems:

- Developers have to learn multiple programming models.
- Code developed for one type of interface typically cannot be used for the other type of interface.

- It is uncommon to have both local and web-based user interfaces for an application, even though this could result in a better user experience for local users. Usually, it's simply too expensive to implement two interface tiers.

2.2.3.3.2 No automatic state management

Developers using VB6 forms and local components are accustomed to making the user interface more convenient by creating forms that remember things for the user – the interface maintains state. If a piece of information is placed in a text box, it stays there until it is explicitly changed or removed by the developer or user.

ASP, however, has no such capability. Every time a page is rendered, one must make sure that all the visual controls have their information loaded. It is the programmer's responsibility to manage the state in the user interface, and to transfer state information between pages.

This means that developers have to write a lot of code for Internet user interfaces that is not relevant to the business problem the application is designed to solve. In addition, if an Internet application is going to run on a group of web servers (often called a web farm), then considerable additional work is necessary to design a state management system that is independent of a particular server.

2.2.3.3.3 Weak user interfaces over the web

It is possible to produce sophisticated user interfaces for the Web by using DHTML and writing a lot of JavaScript. However, most web-based applications actually offer fairly primitive user interfaces because it takes too much time and expertise to write a sophisticated one. (Including a lot of nice graphics doesn't make a user interface sophisticated – it just makes it pretty.)

Developers who cut their teeth on producing state-of-the-art interactive user interfaces in VB during the mid-1990s were never satisfied with the compromises necessary for web interfaces. Better user interfaces on the Web would be an enormous boost for user productivity.

2.2.3.3.4 The need to abstract the operating system

Today's applications need to use the Windows API for a variety of purposes. VB6 developers use the API to monitor Windows messages, manipulate controls, read, and write INI files, and a variety of other tasks.

This is some of the fussiest programming VB6 developers ever have to do. The Windows API is hard program to for a variety of reasons. It isn't object-based, which means we must learn complex calls to functions with long lists of arguments. The naming scheme for the functions is inconsistent and since the whole API is written in C/C++, getting calling conventions right on data types such as strings is very messy.

There's a larger issue here as well. As hardware platforms proliferate, it's no longer enough for software just to run on desktop clients and servers. There are handheld and wireless devices of various kinds, kiosks, and other types of systems, many of which run on different processors and don't use standard Windows as an operating system. Any software written with calls to the Windows API won't be portable to any of these systems without major changes. The only way that software produced with Microsoft tools can become more portable is to abstract away the Windows API, so that application software does not write directly to it. This actually creates the possibility of an equivalent layer of abstraction on other platforms that could allow Microsoft-based software to run on them.

All of these limitations had to be addressed, but Microsoft decided to look beyond just Visual Basic and solve these problems on a global level. All of these limitations are solved in Visual Basic .NET (VB.NET) through the .NET Framework.

2.2.3.4 The solution – Microsoft .NET

Microsoft's .NET initiative is broad-based and very ambitious. It includes the .NET Framework, which encompasses the languages and execution platform, plus extensive class libraries providing rich built-in functionality (Microsoft, 2003). Besides the core .NET Framework, the .NET initiative includes protocols (such as the Simple Object Access Protocol, commonly known as SOAP) to provide a new level of integration of software over the Internet, and a set of pre-built web-based services called .NET My Services (formerly codenamed Hailstorm).

Microsoft also released several products early in 2001, which were described as being part of the .NET Enterprise Server family: SQL Server 2000, Commerce Server 2000, BizTalk Server, Exchange 2000, Host Integration Server (the successor to SNA Server), and Internet Security and Administration (ISA) Server (the successor to Proxy Server).

Some of the marketing literature for these products emphasizes that they are part of Microsoft's .NET strategy. However, it is important to understand the difference between these products and the .NET Framework. The .NET Enterprise Servers are *not* based on the .NET Framework. Most of them are successors to previous server-based products, and they use the same COM/COM+ technologies as their predecessors.

These .NET Enterprise Servers still have a major role to play in future software development projects. When actual .NET Framework projects are developed, most will depend on the technologies in the .NET Enterprise Servers for functions like data storage and messaging.

2.2.3.4.1 General goals of .NET

- **Creating Highly Distributed Applications:** The trend in business applications is towards a more highly distributed model. The next generation of applications may have their elements distributed among various organizations. This contrasts with today's dominant model in which all the elements of an application (except possibly a browser-based client) are located solely within a single organization (Microsoft, 2003).

- **Simplifying Software Development:** Developers need to be able to concentrate on the business logic in their applications, and to stop writing logic for things like state management and scalability. Writing software for the Internet should not require expertise in a long list of Internet-specific technologies. A related goal is to have development for the Internet look very much like development for other platforms. A component accessed over a local network or over the Internet should be manipulated with code very much like that for a component accessed on the local machine. The software platform should be able to take care of the details in transmitting information to and from the component (Microsoft, 2003).
- **Better User Interfaces over the Web:** User interface development also needs to be as similar as possible for the Internet compared to local access. While using local, platform-specific interfaces will always offer more flexibility than a browser-based interface, Microsoft .NET aims to make those two types of interfaces as similar to develop as possible. By making web-based user interfaces richer and more flexible than they are now, bringing them as close as possible to the richness of local, forms-based interfaces (Microsoft, 2003).
- **Simplifying Deployment:** The problems of DLL Hell and the need for large installations of form-based applications are just two examples of current deployment issues. Microsoft .NET aims to make deployment as simple as it was for DOS – just copy a compiled module over to a system and run it. No registration, no GUIDs, no special installation procedure (Microsoft, 2003).
- **Support for a Variety of Languages:** While the idea of one grand, unifying language sounds good in theory, in the real world, different types of developers need different tools. Microsoft .NET is designed to support a multitude of languages, from Microsoft and third-parties. This will allow the development community to evolve languages that best fit various development needs (Microsoft, 2003).
- **An Extendable Platform for the Future:** A new platform needs the capability to adapt to changing conditions through extensions and variations. .NET is

designed with greater extendibility and flexibility than any previous software development platform (Microsoft, 2003).

- **Future Portability of Compiled Applications:** Operating systems will make major changes and perhaps entirely new ones will be introduced in the future. Investments in software development need to be carried forward to those platforms. The goal of .NET is to allow applications to move from current platforms to future platforms, such as 64-bit operating systems with a simple copy, and no recompilation (Microsoft, 2003).

2.2.3.4.2 The structure of Microsoft .NET

To understand how the general goals of Microsoft .NET are accomplished, the general structure of Microsoft .NET should be understood. One way to look at .NET is to see how it fits into the rest of the computing world. Here is a diagram of the major layers of .NET, showing how they sit on top of an operating system, and provide various ways to interface to the outside world. The entire architecture has been created to make it as easy to develop Internet applications as shown in Figure 2.34.

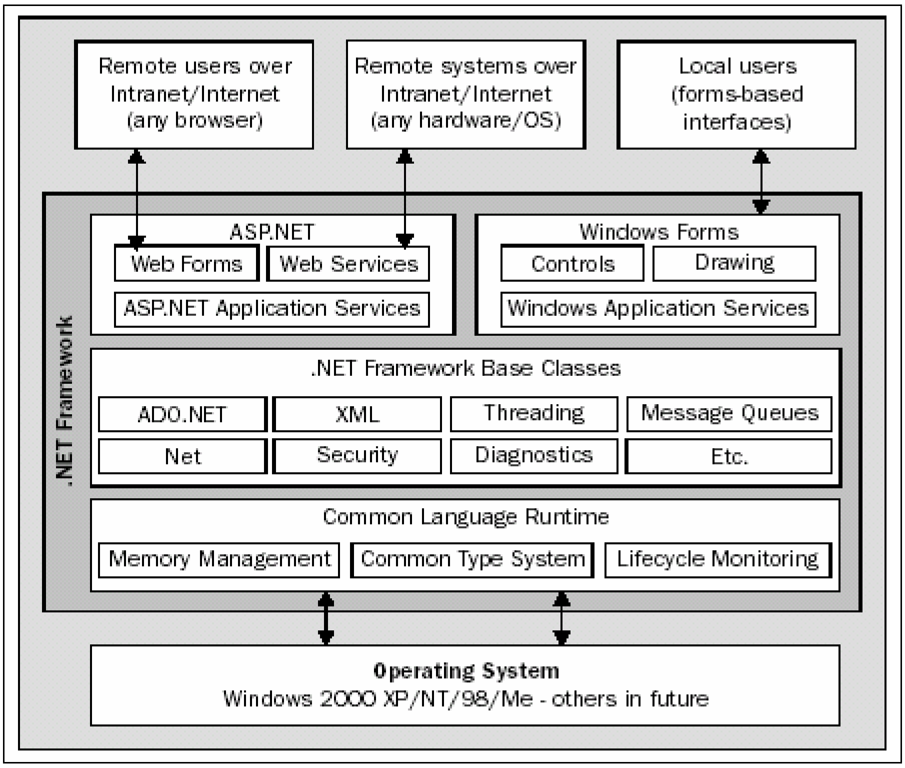


Figure 2.34 Microsoft .NET framework

The first point of the above diagram is that .NET is a framework that covers all the layers of software development above the operating system. It provides the richest level of integration among presentation technologies, component technologies, and data technologies ever seen on a Microsoft, or perhaps any, platform.

The .NET Framework wraps the operating system, insulating software developed with .NET from most operating system specifics such as file handling and memory allocation.

The .NET Framework itself starts with the execution engine, memory management, and component loading, and goes all the way up to multiple ways of rendering user and program interfaces. In between, there are layers that provide just about any system-level capability that a developer would need.

2.2.3.4.3 N-tier application in Microsoft .NET

N-tier applications have become the norm for building enterprise software today. To most people, an N-tier application is anything that is divided into discrete logical parts. The most common choice is a three-part breakdown—presentation, business logic, and data—although other possibilities exist. N-tier applications first emerged as a way of solving some of the problems associated with traditional client/server applications, but with the arrival of the Web, this architecture has come to dominate a new development.

The Microsoft Windows® DNA technology has been a very successful foundation for N-tier applications. The Microsoft .NET Framework also provides a solid platform for building N-tier applications. Yet the changes .NET brings should make architects re-think some of the lessons they've learned in the Windows DNA world about designing N-tier applications. Even more important, the fundamental support for XML Web services built into the .NET Framework allows building new kinds of applications that go beyond the traditional N-tier approach. Understanding how best to architect applications for .NET requires knowing what changes in this new world and how to exploit these changes. N-Tier logical model of Microsoft .NET is given in Figure 2.35. The various layers, which exist in Microsoft .NET framework, are explained below:

- **The Data Layer:** The data layer can usually be split into two separate layers. The first will consist of the set of stored procedures implemented directly within the database. These stored procedures will run on the server and provide basic data only. Not only are they pre-compiled and pre-optimized, but they can also be tested separately and, in the case of SQL Server 2000, run within the Query Analyzer to make sure that there are no unnecessary table scans. They should be kept as simple as possible and cursors or transactions should not be used. Cursors are slow because processing rows one by one instead of as a set is inefficient. Transactions will be handled by the layer above, as ADO.NET gives us much more control over these things.

The next layer consists of a set of classes which call and handle the stored procedures. One class per group of stored procedures will be needed which will handle all Select, Insert, Update, and Delete operations on the database. Each class should follow OO design rules and be the result of a single abstraction - in other words handle a single table or set of related tables. These classes will handle all requests to or from the actual database and provide a shield to the application data. All requests must pass through this layer and all concurrency issues can and must be handled here. In this way data integrity is maintained and that no other source can modify the data in any way.

If the database is being changed for any reason, the data layer can be easily modified to handle them without affecting any other layers. This considerably simplifies maintenance.

- **Business Rule Layer:** This layer is implemented in order to encapsulate the business rules.

  Here classes can be found which implement the business functionality. They neither access data (except through the data layer) nor do they bother with the display or presentation of this data to the user. All should be interested in at this point are the complexities of the business itself. By isolating this functionality, we are able to concentrate on the guts of our system without the worry of design, workflow, or database access and related concurrency problems. If the business changes, only the business layer is affected, again considerably simplifying future maintenance and/or enhancements.

  In more complex cases it is entirely possible to have several business layers, each refining the layer beneath, but that depends on the requirements of the system.

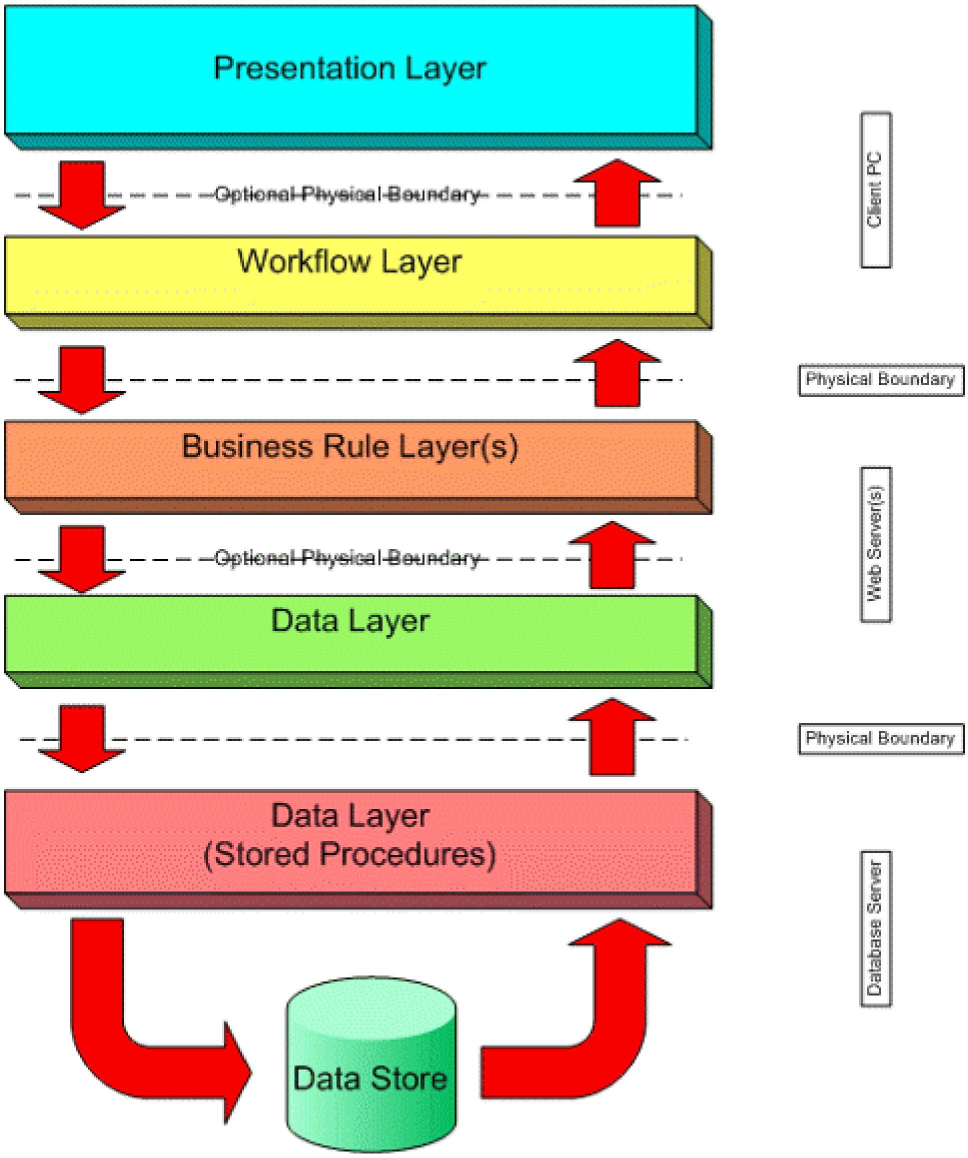


Figure 2.35 N-tiers logical model

- **Workflow Layer:** This is one of the optional layers and deals with data flow to and from your system. It may or may not interact directly with the user interface, but always deals with external data sources.

  For instance, if messages are sent or received from a messaging queue, a web service should be used for extra information, information should be sent or received to another system, the code to handle this would be in this layer. It can be wished to wrap the whole application in XML so that the choice of presentation layer can be expanded. This would also be handled in the Workflow Layer.

- **Presentation Layer:** This layer handles everything to do with the presentation of the system. This does not just include the windows or web forms (or user interface), but also all the classes which will help to present the data.

  Ideally, the event method implementations within the form classes will only contain calls to the presentation layer classes. The web or windows forms, used for visual representation only interface seamlessly with the presentation layer classes which handle all translation between the business layer/workflow layer and the forms themselves. This means that any changes on a visual front can be implemented easily and cheaply.

  Each section (or layer) of the application is a standalone entity which can communicate with layers above and below it. Each layer is designed independently and protected from the others by creating extensible interfaces. All changes can therefore be encapsulated within the layer and, if not major, will not necessarily affect layers above and below it.

CHAPTER 3

LITERATURE REVIEW OF RELATED RESEARCH WORK

3.1 New Business Challenges

One of the basic assumptions of this research is that the virtual enterprise is relevant among other things derived from changes in the market and improvements in technology and infrastructure. However, not to let this assumption stand alone, this section will describe what is seen as a background theory. That is, the theory that is not considered as the principal theory in relation to the described research question, but theory that can help elucidate the need for why the virtual enterprise model can be considered relevant in today's business environment, and as such justify the succeeding research questions addressing the 'what' and 'how' of virtual enterprises. Therefore, before addressing the research questions, this section will address the background question: *Why is the Virtual Enterprise relevant?*

The intention is not to make a thorough examination of all relevant theory addressing the question, but to make probable that companies in fact are faced with new business challenges for which the virtual enterprise could be seen as a possible solution. As already mentioned; no new knowledge will be revealed in relation to the background question. The intention is solely to establish without reasonable doubt that enterprises are facing the listed challenges.

3.1.1 New challenges

In today's business environment corporations are faced with increasing requirements in terms of speed, quality, price, added value, and customer tailored products. The speed of change is increasing due to enhancements and innovations in infrastructure, transportation, technology and hereunder especially information- and communication technology (ICT). Davidow and Malone, the authors behind the book 'The Virtual Corporation', address the pace of the change within ICT:

> *"in forty years computing has experienced a combined improvement in five dimensions – mass storage, reliability, cost, power consumption, and processing speed – of thirty orders of magnitude… It is equal to the jump from the diameter of a single atom to that of the Milky Way galaxy."* (Davidow & Malone, 1992).

In comparison Davidow & Malone state that it took only a change of two orders of magnitude to spark the Industrial Revolution (with the construction of the first textile machines and the first general industrial usage of steam power) and only four orders of magnitude in explosive power to end a world war and redirect human history (with the development of the atom bomb) (Davidow & Malone, 1992).

This change put the traditional stable business model under pressure in more and more industries. Throughout the literature the need for more dynamic enterprise forms are discussed (Malone & Laubacher, 1998; Hammel, 1999; Filos & Banahan, 2001; Mowshowitz, 2001). To quote a few:

> *"We are in the midst of a new business revolution"* (Davidow & Malone, 1992)
> *"It is widely recognized that we are in the midst of an organizational revolution. Throughout the 1980s, organizations around the world responded to an increasingly competitive global business environment by moving away from centrally coordinated, multi-level hierarchies and toward a variety of more flexible structures that closely resembled networks rather than traditional pyramids"* (Miles & Snow, 1992)
> *"If you think the information revolution isn't transforming your business, think again."* (Rayport & Sviokla, 1994)
> *"For many firms in the twenty-first century, success will come from their ability to continuously create new products and services in an expanding global economy.* (Oates, 1998)

“One, the down side, neither the organizational model essential to facilitate continuous innovation, nor the business model necessary to turn its value-adding potential into profits, has been fully articulated." (Miles et al, 2000).
“In the same way that Newton’s view of the world had to give way to Einstein’s relativity we too need to replace our 18th century views of business to make way for the advent of new models for understanding the interactions of markets.” (XEconomy, 2001)

In overall general terms the evolution can be illustrated as shown in Figure 3.1. That is, manufacturing have moved from craft production within one organization; through mass production; a focus on core competencies; and towards customization where a large portion of the value is added outside the organization. As a part of this evolution, the focus has shifted from “production cost” to “production rationalization” and lately towards “customer satisfaction” as shown on the figure (Jagdev & Thoben, 2001). Production rationalization is to produce what the customer seeks - for a price he/she will buy it from you. Please note that often (mass) customization is related to utilization of mass production e.g. through inclusion of mass-produced components in customized products.

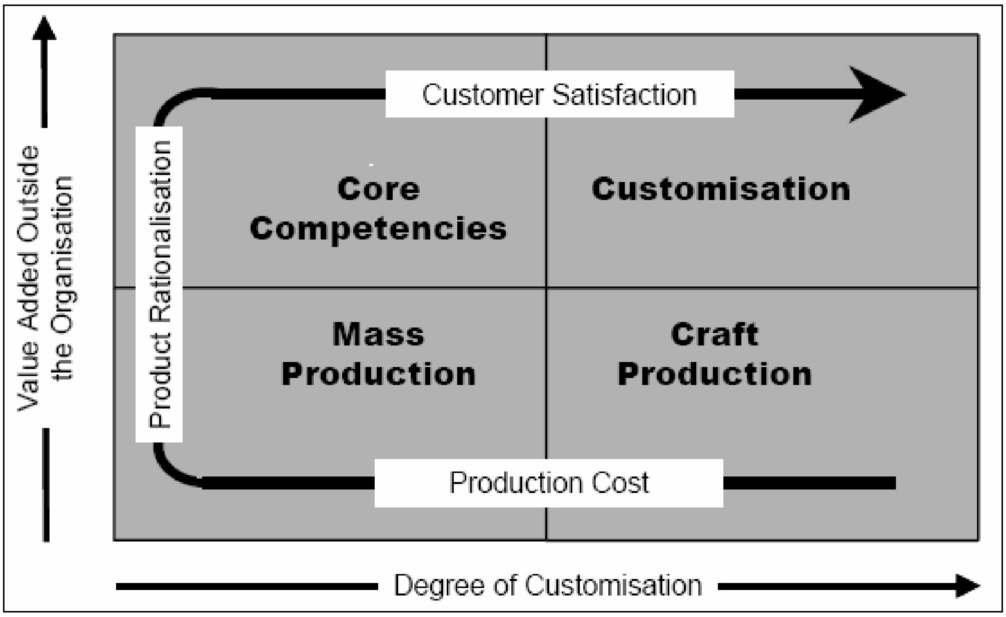


Figure 3.1 Craft production through mass production to customization

In general four interrelated major ‘forces’ put additional requirements on today’s enterprises:

- Globalization;
- Focus on core competencies through outsourcing;
- Customization;
- Impact from the development in technologies.

As mentioned these force are not independent. For instance: the development in technologies enables companies to become more global. The globalization, forces the competition, which pressures companies to focus on their core competencies, which again enables them to provide more customized solution to the market through networking and collaboration.

The four ‘forces’ will be described separately in the following sections.

3.1.2 Globalization

The statement “the business-world is becoming smaller” although a cliché often heard holds much truth. Enterprises co-operate more and more extensively with other enterprises in all the phases of a product’s life cycle:

> *"Global competition is pervasive. From semiconductors to cleaning services, no industry is free from the impact of global competition… Globalization has changed the boundaries of competition"* (Prahalad & Hamel, 1994)
> *"Manufacturing now takes place in a global economy"* (Browne et al, 1995)
> *“All institutions have to make global competitiveness a strategic goal. No institution, whether a business, a university or a hospital, can hope to survive, let alone to succeed, unless it measures up to the standards set by the leaders in its field, anyplace in the world”* (Drucker, 1999).
> *"Today’s manufacturing systems are subject to tremendous pressures because of the ever changing market environments and the fact that manufacturing*

takes place in a global context where local markets are subject to global standards" (Jagdev & Thoben, 2001)

Thus, organizations need to expand their scope to become global, and differentiate their patterns of cooperation to encompass collaborative activities. One of the implications of the globalization is that it is no longer possible to base a business or a country's economic development on cheap labor (Drucker, 1999).

"a business – except for the smallest and most purely local one, for example, a local restaurant – is unlikely to survive, let alone to prosper, unless its workforce rapidly attains the productivity of the leaders of the industry anyplace in the world. This is true particularly in manufacturing." (Drucker, 1999)

Therefore, to survive in the global world companies have to be among the best in their particular field. This however is not new, but the globalization makes the competition more global and thus fiercer than earlier. As a result companies have to concentrate on their core competencies in order to face the increased competition derived from the globalization, cf. the following section.

3.1.3 Core competencies and outsourcing

Outsourcing and a focus on core competencies – among other things to allow for economies of scale in operations including knowledge work such as engineering – requires better collaboration, synchronization of processes, and appropriate handling of time and distance constraints. The increased competition derived from among other things globalization (ref. section 3.1.2) has made more companies focusing on their core business and outsourcing business that are not vital for them or business in which they do not posses competitive resources. In some situations it even makes companies collaborate with their competitors and winning through that (Hammel et al, 1989).

Outsourcing is about letting other companies perform activities that were previously performed within the company. It is not only about outsourcing periphery activities and processes but also processes that the company does not master sufficiently:

> *"In a virtual enterprise, specialist firms perform critical processes – not because there're unimportant but rather because they're so important that the original company can't afford to have them handled in a mediocre way."* (Hammer, 2000)

The notions of resources, capabilities and competencies have gained considerable attention from scholars of strategic management research as they attempt to explain why companies differ in organizational performance (Pedersen & Berg, 2001).

The resource-based view (RBV) of the firm was introduced by (Wernerfelt, 1984). However when the paper appeared in 1984 it was ignored. It was not until Prahalad's and Hamel's article in *Harvard Business Review* in 1990 (Prahalad & Hamel, 1990) that the RBV was diffused into practice (Wernerfelt, 1984).

The idea of looking at firms as a broader set of resources originates, according to Wernerfelt, back to the work of Penrose in 1959, but the work had up to that time received relatively little attention (Wernerfelt, 1984).

Before the resource-based view the traditional perspective when formulating strategies of an enterprise was to look at the products (Wernerfelt, 1984). In the RBV the perspective changes towards looking at the resource of the enterprise when choosing e.g. which (diversification) strategy to select.

Thus, a company is not diversified by its products; it is through its competencies (which it can turn into products) that a company can achieve sustainable competitive advantages.

Applying the concept of entry barrier from Porter's five forces (Porter, 1980) Wernerfelt looks at the RBV as a way to establish what he calls *resource position barrier*. That is, the fact that someone already has the resource affects the costs and/or revenues of later acquirers adversely. In these situations the holder of the resource can be said to enjoy the protection of a resource position barrier (Wernerfelt 1984).

Resources, capabilities and competencies are related terms sometime used synonymously in the literature. A distinction between the terms is given in (Pedersen & Berg, 2001). The following is a summary derived from these pages.

Resources are defined as *"stocks of available factors that are owned or controlled by the company"* (Amit & Schoemaker, 1993). Resources encompass both tangible and intangible assets (Wernerfelt, 1984) and can basically be categorized into three types: physical capital resources, human capital resources and organizational capital resources (Barney, 1991). Physical capital resources include the physical technology, facilities, equipment, and materials. Human capital resources include education, experience, judgement, intelligence, and relationships. Organizational capital resources include formal reporting structure, formal and informal planning, controlling and coordinating systems as well as informal relationships (Barney, 1991).

Capabilities refer to a company's capacity, or ability, to deploy resources (Amit & Schoemaker, 1993). Capabilities are information and knowledge-based processes deploying resources in an often coordinated way to affect a desired outcome as for instance products, services, organizational behavior, trust and the like. Like resources, capabilities can be both tangible and intangible, but they are in essence limited to the information and knowledge handling of human capital resources (Amit & Schoemaker, 1993).

Competencies are complementarily a subset of resources and capabilities with the potential to lead to competitive sustainable advantages (Porter, 2001). Whether a competitive advantage is sustainable or not is not defined by a period of calendar time, but on the inability of current and potential competitors to duplicate it (Barney, 1991). Pedersen & Berg list a set of criteria that resources and capabilities must meet in order to constitute a competence:

> "*they must promote company efficiency and effectiveness and be perceived as valuable by the market and environment;*
>
> a) *they should not be readily available at competitors;*
> b) *they should be difficult to acquire for competitors;*
> c) *they should not have strategically equivalent abundant substitutes*
>
> *Note that a competence here is defined relative to competencies of other companies. Besides, a competence should not be confused with a specific product or service, rather a competence spans the ability to create several products or services"* (Pedersen & Berg, 2001).

Another essential characteristic of a competence is that it does not diminish when used; on the contrary they can be enhanced when they are applied and shared (Prahalad & Hamel, 1990).

All in all companies in general have to raise their awareness concerning their (core) competencies. They have to focus on what they do best (and leave the rest to others). Once the competence has been identified it should be nurtured and enhanced on an ongoing basis in order to maintain it as a competence in order to ensure that the competitive advantage can be sustained. Corporations have to look out for applying their competencies in non-traditional ways (Prahalad & Hamel, 1990). That is, to maintain resources and capabilities as competencies and to improve existing competencies even further the possessor of the competencies (i.e. the enterprise) has to look for alternative ways to apply the competencies in new settings and thereby increase the frequency of the competence's use. The management process of securing a broad application (multi-project, multi consortium application) of a competency could be named *competence unfolding* as suggested by (Vesterager et al, 1999).

The focus on the corporation core competencies has lead organizations to downsize and outsource their none-core business. As a part of this transition some companies have disintegrated into smaller parts (Malone & Laubacher, 1998; Oates, 1998).

In 1998, Malone & Laubacher guess about what the future scholars will think when looking back on the changes the current business in undergoing:

> *"In the year 2022, the Harvard Business Review will be celebrating its one hundredth year of publication. As a part of its centennial celebration, it may well publish a series of articles that look back on recent business history and contemplate the massive changes that have taken place. The authors may write about the industrial organization of the twentieth century as merely a transitional structure that flourished for a relatively brief time. They may comment on the speed with which giant companies fragmented into the myriad micro businesses that now dominate the economy. And they may wonder why, at the turn of the century, so few saw it coming."* (Malone & Laubacher, 1998)

Although it is a speculative statement the point is that the current business will be replaced by smaller businesses. The disintegration of larger companies has already affected some of the largest companies in the USA, cf. the following quote:

> *"Twenty-five years ago, one in five U.S. workers was employed by a Fortune 500 company. Today the ratio has dropped to less than one in ten."* (Malone & Laubacher, 1998)

Although indicating that smaller companies will be more dominating it will not necessarily become true. What makes small companies successful is their ability to responding rapidly to changing customer needs. It is not necessarily related to the size of the company:

"Many large companies have drastically downsized, divested, and outsourced to reduce the costs and complexity of their operations. Yet simply reducing the size of a corporation is not the solution As CEO Jack Welch [of General Electrics, ed.] has said, GE's goal is not to become smaller but to 'get that small-company soul and small-company speed inside our big-company body'" (Haeckel & Nolan, 1993).

Thus, the task is not to downsize by any means. The challenge is to assess what to keep in relation to the company's competencies and then through that achieve the needed flexibility and responsiveness that can make the company an attractive player in the market.

Thus, in prolongation of – and to some extent as an answer to - the globalization trends described in section 3.1.2 companies are undergoing a transition towards a focus on their core competencies and at the same time outsource their non-core activities. Companies that have developed and nurtured competencies that can outperform or at least match those of their competitors have established a foundation for achieving sustainable competitive advantages.

It is however, not sufficient to master a certain competence to perfection. As mentioned in the criteria above the competence has to be perceived as valuable by the market and the environment. In more situations this requires collaboration with other partners possessing complementary competencies, and with whom valuable solutions can be created. This type of collaboration could for instance be in the form of a virtual enterprise. In terms of creating valuable solution to the market a higher level of customizations is required as will be described in the next section.

3.1.4 Customizations

In addition to the globalization trend and the ongoing movement towards focusing on core competencies described above, companies have to adapt to an increasingly demanding market requiring more customized solutions:

"Rather than follow the make-and-sell strategy of industrial-age giants, today's successful companies focus on sensing and responding to rapidly changing customer needs." (Haeckel & Nolan, 1993)

The higher customization level is a consequence of the increased competition, derived from the described increased globalization and focus on core competencies, as well facilitated by the development in ICT:

"Information processing, networks, databases, and expert systems software combined with various special-purpose data-collection, data-recording, and data- handling equipment makes it possible in many businesses to treat customers as individuals." (Goldman et al, 1995)

Thus in terms of customization there has been quite a shift from the first days of mass production by Henry Ford:

"People can have the Model T in any color -so long as it's black."

The accepted philosophy in mass production was, "If we can make it, they will buy it." (Goldman et al, 1995) To compete in the future companies would have to make their products more like solutions addressing individualized customer needs (Goldman et al, 1995; Hammel, 1996). The customization level of Henry Ford (or lack hereof) is in most cases not sufficient in the present business environment. The customization level needs to be higher and in some situation it needs to move towards the *virtual product* as defined by Davidow and Malone.

"*The ideal virtual product or service is one that is produced instantaneously and customized in response to customer demand.*" (Davidow & Malone, 1992)

Although Davidow and Malone acknowledges that the perfect virtual product can never exist, they claim that there is little doubt that many will come close (Davidow & Malone, 1992).

Treating customers as individuals can help establishing a long-term relationship with the customers, getting inside knowledge about their preferences and future needs. If companies do not manage to establish individual relationships with their customer they face the risk of loosing them to a competitor. ICT makes the cost of switching lower for a customer (Porter, 2001).

When dealing with customers and their expected lifetime values, the companies should also consider the risk associated with a given customer(s). Hence, the company should consider composing a portfolio of customers to minimize long-term risk and to optimize their long-term earning (Dhar & Glazer, 2003). In terms of the aforementioned competencies this means that companies need to find new way of deploying their competencies in new constellations as a way to broaden the portfolio of end-products in which their competencies can contribute to.

The aforementioned globalization and focus on core competencies can be seen as developments that enable and to some extent force companies to provide more customized solutions to their customers. The customization, globalization, and to some extent the outsourcing and accompanying focus on core competencies trends are all facilitated by the development within technology. The technology impact will be described in the following.

3.1.5 Technology impact

Many of the above trends are facilitated and/or supported by the technological development in general and within ICT in particular.

"Information technology has driven much of this dramatic shift by vastly reducing the constraints imposed by time and space in acquiring, interpreting, and acting on information." (Haeckel & Nolan, 1993)

As can be seen in the following quote ICT should be seen as the main enabler for the emerging business models such as the virtual enterprise:

"There is nothing particularly novel about industrial collaboration. Collaboration of some kind or another has existed ever since the industrial revolution. However, classification of the range and scope of cooperations has evolved with the advent of ICT tools. These tools have given manufacturers new means of sharing information and managing operations, thus blurring the organizational boundaries with suppliers, clients and even erstwhile competitors. The quality and efficiency issues extend well beyond the traditional enterprise boundary. Therefore, when one talks about the emerging manufacturing typologies (e.g. Extended and Virtual enterprises) and business paradigms (e.g. EC, i2i, b2b, and so on), one must not forget that the key necessary component in these new labels is the modern ICT tools. Without these ICT tools, these new labels have no meaning." (Jagdev & Thoben, 2001)

The impacts of new technologies such as the Internet are expected to make companies reconsider what they are doing and not least the way that they are doing it. This includes also the concept of a company:

"One of the most important applications of the Internet is flying under the media's collective radar. The virtual enterprise (also known as virtual integration) is changing the entire concept of what it means to be a company." (Hammer, 2000)

The developments within ICT have introduced a new type of market, i.e. the electronic market or *marketspace* (Rayport & Sviokla, 1994). Companies are under a transition from a marketplace towards a combined marketplace and a marketspace. As a result of the development within ICT:

> *"Every business today competes in two worlds: a physical world of resources that managers can see and touch and a virtual world made of information. […] Executive must pay attention to how their companies create value in both the physical world and the virtual world." (*Rayport & Sviokla, 1995)

In the traditional marketplace value is usually created through an aggregation of infrastructure, context and content. For example, a newspaper is an aggregated collection of content (news, business, sports, weather, as well as other information), context (format, organization, logo, editorial style, and rhetorical tone), and infrastructure (printing plant and physical distribution system, including trucks, door-to-door delivery, as well as newsstands sales) (Rayport & Sviokla, 1994). The development within ICT enables enterprises to add or alters content, changes the context of the interactions, and enables the delivery of varied content and a variety of contexts over different infrastructures. With regards to the aforementioned newspaper, this means for instance that in the marketspace the content can be delivered by various news agencies and newspapers, the context can be established by internet portals such as 'my.MSN.com', 'my.Netscape.com' or 'AOL' providing an intelligent front-end to the consumer that enables them to select the content they wish to read. Likewise, the infrastructure has been disaggregated, and has become a means to value creation as well. That is, the infrastructure consist of e.g. telephone lines or network, the connection provided by an Internet provider, and a PC which again can be owned by the consumer or e.g. rented at an Internet cafe.

As a result the emergence of the marketspace enables companies to do things differently, which opens up opportunities to create new forms of value for their customers. On the other hand, it can also enforce the competition from rivals, which again put the average profitability under pressure:

> *"If average profitability is under pressure in many industries influenced by the Internet, it becomes all the more important for individual companies to set themselves apart from the pack."* (Porter, 2001)

Successfully, setting a company apart from the pack is a way for companies to become more profitable than the average performer. This of course it is not reserved to Internet related business; in general all commercial companies strive to be more profitable.

A competitive advantage can be hard to keep sustainable purely by improving operational effectiveness. Once a new best practice is established competitors will quickly strive to copy it. Effectiveness derived from the open platforms and common standards such as the Internet can be hard to keep. Porter claims that simply improving operational effectiveness does not provide a competitive advantage:

> *"The only way to generate higher levels of economic value is to gain a cost advantage or price premium by competing in a distinctive way."* (Porter, 2001)

As is often the case, when dealing with ICT and its influence on how companies do business it is not a question of either or but more a matter of getting the right mix of 'bricks' and 'clicks' (Gulati & Garino, 2000). Virtual activities do not eliminate the need for physical activities, but often amplify their importance (Porter, 2001).

The development within ICT does not make the traditional value chain obsolete. The possibilities derived from the ICT and hereunder the Internet should be integrated in the traditional business specifying a new overall strategy for the company:

> *"By creating separate Internet strategies instead of integrating the Internet into an overall strategy, companies failed to capitalize on their traditional assets, reinforced me-too competition, and accelerated competitive convergence."* (Porter, 2001)

Although there is no doubt that the development within ICT has influenced the business of many people, it is not revolutionizing everything:

> *"In periods of transition such as the one we have been going through, it often appears as if there are new rules of competition. But as market forces play out, as they are now, the old rules regain their currency. The creation of true economic value once again becomes the final arbiter of business success."* (Porter, 2001)

> *"The 'new economy' appears less like a new economy than like an old economy that has access to a new technology… In our quest to see how the Internet is different, we have failed to see how the Internet is the same."* (Porter, 2001)

Thus, although new technologies are being developed all the time it is not changing that fact that classical virtues of good business is still valid. ICT should be seen as an enabler that broadens the market, facilitates communication, and collaboration. Having said that the ICT should not be applied uncritically, companies should re-consider the appropriateness of how they do business and modify it in accordance to them possibilities emerging from the technology challenging among other things their old business habits. That is:

> *"Use computers to redesign – not just automate existing business processes."* (Hammer, 1990)

This is supported by Davenport and Short who argue that there is a recursive relationship between emerging ICT capabilities and business process re-engineering (Davenport & Short, 1990). That is:

> *"Thinking about information technology should be in terms of how it supports new or redesigned business processes, rather than business functions or other organizational entities. And business processes and process improvements*

should be considered in terms of the capabilities information technology can provide." (Davenport & Short, 1990)

Conclusively, the ICT developments pressure enterprises to focus on what they are good at and continuously improve it utilizing the latest technology. And at the same time let partners do the rest. One of the consequences of the development within ICT is that new business structures utilizing the new technology are needed:

> *"The winners in a civilization remade around computers and the Internet will not be those who attempt to contain the technology. The winners will be those who invent new structures of government, business, and society in which technology is embedded."* (Davidow, 2000)

Another, very recent comment to the fuzz regarding the impact of technology is given in the October 2003 issue of Harvard Business Review:

> *"Yes, something big did happen in the 1990s. But it was less about technology and more about new forms of competition."* (Farrell, 2003)

New forms of competitions in the form of emerging new business structures will be addressed in the upcoming section 3.2.

3.1.6 Summary of new business challenges

Companies are facing new requirements derived from the globalization trend, outsourcing and a focus on core competencies, customization, and impact from technology. Although these factors have been addressed separately above, the four forces, as mentioned in the introduction, are not independent from each other. In general the technology improvements and hereunder development within ICT can be seen as an enabler for the other three. It has reduced some of the barriers towards globalization for instance by enabling people to communicate and interchange electronic documents and models on-line. The technological developments have also reduced the cost and time of transporting goods as well as people from one place to another. In addition to – or to some extent caused by – the globalization and the competition derived from it, companies are focusing on their core competencies and outsourcing their non-core business to other partners.

The globalization makes the competition fiercer within many industries. Along with the technological development it - other things equal - reduces the entrance barriers to the market for competitors. As a consequence companies are forced to reconsider the scope of their business and focusing on what they do best. This means identifying and nurturing their core competencies and outsourcing none-core activities to external partners. An effect from the globalization and the focus on core competencies is customization. The customization trend can be seen as both enabled by as well as triggered by the globalization and the companies focusing on their core competencies. On one hand companies internal quest for mastering their own competencies enables companies to offer new and more customized solutions. On the other hand the increased competition derived from the globalization introduces new customized solutions to the market, which again forces companies to develop more customized solutions themselves.

Customization combined with the globalization trends and the focus on core competencies put additional requirements on enterprise. All in all this means that companies to a larger extent have to be global, focus on their core competencies and partner up with companies possessing competencies that complement own competences and with whom customized solutions can be made within a competitive ‘time to market’.

These new requirements put the traditional stable business model under pressure in more and more industries. As a result new more dynamic business models are emerging such as virtual enterprises. These new enterprise forms will be addressed in the following sections.

3.2 Emergence of new organizational concepts

Given the described new business challenges the traditional principles for enterprise management, which rely on and reflect a situation with more local and stable markets and relationships, will no longer be sufficient. This has been emphasized by many authors. An example from 1984:

> *"We expect the 21st century firm to be a temporary organization, brought together by an entrepreneur with the aid of brokers and maintained by a network of contractual ties."* (Miles & Snow, 1984)

Different types of new organizational forms have been addressed in the literature. These new organizational models for manufacturing enterprises are listed in below:

- Biological Manufacturing
- Fractal Company
- Lean Production
- Agile Manufacturing
- Holonic Manufacturing
- Virtual Enterprise
- Extended Enterprise
- Breeding Environment

In the following sections the focus will be on the literature related to these new organizational concepts. The literature will be addressed in the context of the first research question outlined in section 1.2.2, i.e. RQ1: "*What is a virtual enterprise?*"

3.2.1 Biological manufacturing

Biological manufacturing (or bionic manufacturing as it has more recently come to be known) is a design methodology for manufacturing systems that imitates biological construction (self organization) and optimization (evolution) and is based on cooperative, flexible, and adaptive individual production units which can individually "evolve" to achieve an overall manufacturing strategy. Taken as a whole, the manufacturing system is seen as an individual organism that responds to environmental stimuli by manufacturing products. These products are also viewed as individual organisms that compete against other products in characteristics such as shape, function, cost, etc. This model of manufacturing as a biological system was worked on by Kanji Ueda at the Information Processing Center of Kobe University in Japan. The basic tenets of the approach are discussed in (Ueda, 1992) and entail a shift away from centralized control with emphasis on what Ueda refers to as "interactive" manufacturing.

In such a model, the designer, manufacturer, and consumer "interact" throughout a product life cycle as opposed to a more traditional approach that views the life cycle sequence as a linear process of minimally related stages. Robert W. Galvin, Chairman of the Executive Committee of the Board of Directors for Motorola has stated that,

> *"…as global companies grow to mammoth proportions, the only way they will be managed effectively is by using a biological model that mimics the self-governing and learning techniques of a complex organism…"* (McCormack, 1997)

In general, bionic manufacturing structures mimic cellular activity with characteristics such as the static, genetic information found in deoxyribonucleic acid (DNA), or the adaptive, learned information found in brain neurons. Tharumarajah et al. highlight the similarities between organic cellular systems and manufacturing processes stating that:

> *"A biological viewpoint has close parallels in manufacturing. For example, production units on the shop floor can be compared to cells in biology. Biologically, a cell is separated from outside by a membrane through which material enters and exits. In a cell there is an organelle, which creates the cell's functions. Cells are immersed in an environment which is chemical.*

Substances are taken in from this environment. There is also an internal environment and the cell operations exchange (chemical) information with both inside and outside environments. A cell changes its own conditions by its operations and it can do multiple and different operations. These properties correspond closely with autonomously operating manufacturing units… These units obtain the needed inputs from the factory floor and perform operations. Outputs of these operations flow back to the environment. The operations require information from both internal and external environments." (Tharumarajah et al, 1996)

3.2.2 Fractal company

The concept of a fractal company has its origin from a book written by Hans-Jurgen Warnecke entitled, *The Fractal Company: A Revolution in Corporate Culture*. In the book, (Warnecke, 1993) proposes an organizational strategy through which a company can be created from small components, or "fractal objects," that have the ability to adapt quickly and reorganize in the face of a rapidly changing market environment. The key characteristics of these objects include:

- Self-organization - there is no functional intervention required by a higher authority for operation or regeneration.
- Self-similarity - each object in a fractal firm is similar to all others, supporting similar components and sharing similar goals.
- Self-optimization - each object performs continuous internal self-improvement.

The inspiration for the term fractal organization arises from the perception of production process and control as an interacting system of atomistic components. Venkatadri et al. discuss the implementation of fractal concepts in a job shop production environment. They describe their work as an alternative to the agile manufacturing concept [agile manufacturing is discussed in the Section 3.2.4], stating that:

"It was in this context that the fractal layout was originally conceived, intended to be an agile manufacturing alternative achieved through the creation of multifunction mini factories within the confines of a factory." (Venkatadri et al, 1997)

Spivey et al. observe that the success of a fractal solution is dependent upon a comprehensive top-to-bottom implementation such as that espoused by the business process re-engineering concept. Spivey et al state that:

"…each of these factors cascades into several interrelated sets of concerns. For example, the management factors comprise concerns about leadership and the management system. The resource factors include concerns about information infrastructure, time, and money." (Spivey et al, 1997)

"Regardless of the level of detail at which the framework is viewed, improving the NPD [ed. new product development] process requires attention to all of these factors, by all levels within the organization. For example, visionary leadership on the part of senior management will have little effect if middle management and line supervisors fail to provide the necessary leadership for their respective groups of subordinates." (Spivey et al, 1997)

3.2.3 Lean production

The characteristics of lean production are first identified and encapsulated in a book entitled, *The Machine that Changed the World*, written by James P. Womack, Daniel T. Jones, and Daniel Roos. The lean production concept is based on an earlier five year study of the automobile industry by the International Motor Vehicle Program (IMVP) at the Massachusetts Institute of Technology (Womack et al, 1990). The book presents a cautionary treatise warning that companies in the United States must adopt "lean" production process and practice policies to compete successfully with Japanese companies. This concept is further formalized in a later book entitled, *Lean Thinking: Banish Waste and Create Wealth in Your Corporation* (Womack & Jones, 1996). As implied by the title, lean production is, in its basic form, the manufacture of a product with a minimum of waste. However, the treatise by Womack & Jones takes a broader view of "waste" than just that of material scrap and unnecessary overhead and proposes that a lean implementation address all aspects of value-creating activities.

The concept of lean production represents the natural evolution of "Just in Time" (JIT), a production concept pioneered by Toyota. This idea is to make only what is needed when it is needed. Noori & Mavaddat discuss lean production issues and link them to enterprise wide integration of information and coordination of decisions. Furthermore, they recognize the importance of IT in maintaining competitiveness and company focus, stating that:

> *"The necessity of maintaining lean operations and becoming an agile enterprise in which the speed and flexibility at which a company matches that of its technology is widely accepted. Information technology is providing the means for companies to integrate better their internal and external activities."* (Noori & Mavaddat, 1998)

Furthermore, Plonka discusses the workforce required to support lean and agile enterprises and observes that:

> *"The cognitive demands that accompany technological innovation will require readying the work force for a more demanding environment. Workers have to look beyond loading parts in machines. They will need to be continually involved with the process and to intervene when required. Operators must be familiar with equipment technology in order for them to suggest improvements in the reliability, precision, and serviceability of equipment, controls, and sensors. These demands will require acquisition of new knowledge, accelerated learning, and just-in- time delivery of training.*" (Plonka, 1997)

3.2.4 Agile manufacturing

Agile manufacturing was introduced by Steve L. Goldman, Kenneth Preiss, R. N. Nagel, and Rick Dove in a 1991 report by the Iaccoca Institute at Lehigh University in Bethlehem, Pennsylvania entitled, *21st Century Manufacturing Enterprise Strategy.* The concept presented in the study has since been championed by the Agile Manufacturing Enterprise Forum (now known as the Agility Forum) which has defined the generally accepted long term view of agility in production. Numerous research and development programs are ongoing in this area. A few notable examples include: the Agile Manufacturing Initiative sponsored by the Defense Advanced Research Projects Agency (DARPA) and the National Science Foundation (NSF) which has a mandate to demonstrate and evaluate the processes described in the report; The Technologies Enabling Agile Manufacturing (TEAM) program sponsored by the U.S. Department of Energy with the goal of demonstrating the benefits of integrating multiple software systems within an agile manufacturing enterprise; and the establishment of a series of Agile Manufacturing Research Institutes (AMRI's) which support the teaming of university and industry with the goal of developing the principles and practices that define agile manufacturing (Goranson, 1999).

Despite this high level of interest in agility, the actual definition of the concept is vague and somewhat expansive. Even the Agility Forum notes that the idea of agile manufacturing is not a specified technique with a clearly delineated list of components. As a result, researchers have adopted a wide range of approaches to agility. As an example, Lee defines agility as,

> *"…the ability of a manufacturing system to manufacture a variety of components at a low cost and in a short period of time."* (Lee, 1998)

This approach focuses on interventions early in the design stage of a product, the reduction of lead times for a product, and an increased utilization of resources. On the other hand, (Devor et al, 1997) observe that the agile concept is evolving from an initial localized implementation to become a strategic methodology that utilizes "proactive adaptability". This strategic view results in a more comprehensive, less specific perspective as can be seen in the definition of agility provided by Gunasekaran. He defines agility

> *"…as the capability to survive and prosper in a competitive environment of continuous and unpredictable change by reacting quickly and effectively to changing markets, driven by customer-designed products and services…."* (Gunasekaran, 1998)

Gunasekaran further notes that the concept embodies an inherent paradox as companies must compete and cooperate in the same market environment. In general, the fundamental tenet of agile manufacturing is the use of modern information technology to form virtual enterprises which "agilely" respond to changing market demands. This is accomplished through the formation of a flexible and dynamic manufacturing environment that can utilize a network of manufacturers to produce "customized" products. To accomplish an agile implementation, five elements are proving to be the key (Forsythe, 1997):

- Changes in business, engineering, and production practices
- Seamless information flow from design through production
- Integration of computer and information technologies into all facets of product development and production processes
- Application of communications technologies to enable collaborative work between geographically dispersed product development team members
- Introduction of flexible automation of production processes

Jung et al. describe a strategy for the rapid development of agile manufacturing systems and identify nine architectural elements that must be addressed successfully to achieve agility; control, process, function, information, communication, distribution, development, implementation, and reference (Jung et al, 1996). (Weston, 1998) provides insight into the information (and associated software) infrastructure issues that need to be addressed in an agile implementation and (Reimann & Sarkis, 1996) discuss an organizational framework for achieving agility by linking companies to form a "virtual" enterprise.

3.2.5 Holonic manufacturing

The treatise of holonic manufacturing is founded on concepts presented in 1967 by the Hungarian author and philosopher, Arthur Koestler, who proposed the word "holon" to describe a basic unit of organization in biological and social systems (Koestler, 1967). The holonic manufacturing concept draws on examples in biology in much the same way as the biological (bionic) manufacturing concept. The concept observes that in living organisms and in social organizations entirely self supporting, non-interacting entities do not exist, but rather every identifiable unit of organization is comprised of more basic units while at the same time forming a part of a larger unit of organization. Holon is a combination of the Greek word "holos", meaning whole, and the suffix "on" meaning particle or part. Koestler defined the term, holon, as an identifiable part of a system with a unique identity that is comprised of subordinate parts and is also part of a larger whole.

A holonic approach to design a manufacturing system recognizes that such a system can also be defined as a collection of sub-ordinate parts that in turn are part of a larger whole. This approach recognizes production activities of varying automation and a principle of self organization that does not depend on a hierarchical command structure. Nonetheless, as Sun & Venuvinod point out,

> *"…A holonic organization is not an independent company. Autonomy of a holonic work organization is not absolute and is subject to the requirements of the higher level organization."* (Sun & Venuvinod, 2001)

The strength of holonic organizations is in their ability to support very complex systems that are readily adaptable to change within the modern production environment. The functional and structural concepts of holonic manufacturing support both distributed autonomy and cooperative control (IMS, 2000). The first experimentation with the idea of holonic manufacturing occurred in Japan in the late 1980's. Toshiba and Hitachi, while working independently on unique implementations of the concept, began seeking international partners to expand the research. This attracted the interest of the Japanese Intelligent Manufacturing Systems (IMS) Promotion Center. In 1989, the Center integrated the two programs into a research and development component of the then newly formed IMS Consortium. In 1995, the IMS Consortium was implemented as a full-scale, 10-year research and development program involving Japan, the United States, Australia, Canada, the European Community (EC), and the European Free Trade Association (EFTA) (IMS, 2000). The holonic component of the IMS, known as the Holonic Manufacturing System (HMS) project is comprised of 41 industrial, academic, and research partners.

The goal of holonic manufacturing development is to translate the observations of Koestler into a production implementation that can sustain efficiency, stability, adaptability, and flexibility within a constantly shifting market environment. (Cselenyi and Toth, 1998) discuss fundamentals of holonic production systems and delineate the following holonic "production unit" characteristics:

- Each production unit is economically independent, supporting independent financial management and accounting
- Each production unit is independent in ownership
- Each production unit is cooperative with other production units and is supported by joint services

In an integrated design-to-control approach to holonic manufacturing, Wang states that,

> *"…Based on the holonic concept, the next generation of IMS could form distributed reconfigurable virtual factories, in which human, machines, and control modules interact in dynamically formed virtual clusters. Such a system*

could be built from intelligent, autonomous, and cooperative elements or holons." (Wang et al, 2001)

Along these lines, (Valckenaers et al, 1998) discuss the design of holonic manufacturing systems and identify three holon types; the resource holon, the product holon, and the order holon. The resource holon is a physical resource and the information processing that controls it. The product holon contains the process and product knowledge needed for making a product. The order holon represents a task within the manufacturing process. Using these three holon types, Valckenaers et al. are able to construct a holonic-based manufacturing system model. Bagshaw & Newman take a higher level view and construct an entire small manufacturing enterprise from a three holon system as follows:

> *"Holonic manufacturing represents the enterprise as an enhanced organization holon which can further be categorized into three sub-holons namely: the executive holon that represents the ultimate decision-making process within the company, the business holon that covers administration activities such as order processing, finance, costing, process planning and scheduling etc., and the manufacturing holon involves the execution and monitoring of the process plans produced by the business holon."* (Bangshaw & Newman, 2000)

3.2.6 Comparison of concepts

The advanced manufacturing concepts that have been described in the previous sections (i.e., biological/bionic, fractal, lean, agile, and holonic) are being developed to address the characteristics of a post-2000 market environment. Of the five, leanness and agility tend to focus more on procedural change with supporting organizational change. Agility embraces the utilization of information technology to form virtual enterprises that can "agilely" respond to changing market demands via a flexible, dynamic production environment. In practice, agile manufacturing comprises cooperation among partnered elements with an overall operational unit that is linked by electronic communications channels. The remaining three concepts, biological/bionic, fractal, and holonic, are organizational architectures which define manufacturing systems as a network of distributed, autonomous, flexible production elements. Tharumarajah et al point out that:

> *"Bionic, fractal, and holonic concepts describe, in quite general terms, the underlying principles of designing manufacturing systems which are highly flexible in their structure and operation. These principles advance an organization of distributed autonomous modules that are capable of self organizing behaviors to carry out the necessary functions. However, the concepts differ in their approach to design of these features."* (Tharumarajah et al, 1996)

Each of these concepts is characterized by a networked structure of basic units; for bionic it is the cell, for fractal it is the fractal object, and for holonic it is the holon. A fractal implementation is comprised of self-similar objects (which may differ internally as they self organize) that continuously react to the external environment to achieve a common system of goals. Fractal objects are very dynamic, reconfiguring or even restructuring in response to changes in the environment. A holonic architecture, although similar to a fractal implementation, is instead decomposed by function in which a holon, unrestricted by geographic location, can define anything from an informational element to a specific machine and may have numerous, overlapping "whole-part" relationships. A bionic architecture supports a more layered approach to its cell structure than fractal and holonic implementations. Its evolutionary ability, like a fractal implementation, is unrestricted by functional definition and; therefore, supports a more dynamic reproduction capability than functionally-defined holonic architecture. This capability, unlike a fractal architecture which is based on an initial design, is borne from a natural genesis that passes on basic "DNA" information as each new cell is formed.

3.2.7 Virtual enterprise

As described in section 3.1 the globalization, focus on core competencies, outsourcing, and customization trends as well as the development of new technology are putting pressure on the traditional types of enterprises. As already indicated the virtual enterprise is one of the organizational forms that are emerging from this development.

> *"It is this perpetual transformation, achieving technological leaps about once per decade comparable to the four-thousand-year path from horse cart to bullet train, that makes the virtual corporation inevitable and immediate."* (Davidow & Malone, 1992)

> *"The virtual enterprise truly changes everything. The only question is: When will the mainstream media finally catch on?"* (Hammer, 2000)

> *"There is now strong evidence that we are in the midst of an information revolution that will parallel the agricultural and industrial revolutions that preceded it. Information Technology has previously been used to automate existing working practices. Business organizations now realize that the real gains from deploying technology are not using it to do existing things better, but using it to do things the way they should be done. Possible the most exciting of these new technologies is the development of Virtual Organizations."* (Mazzeschi, 2001)

Some authors such as (Mowshowitz, 2001) claims that the rise of the virtual organizations will not only change the way business is done, it will also call for a fundamental change in governance. It will led to what he call virtual feudalism (Mowshowitz, 2001). How governments should govern however is not within the scope of this research. One could argue that the concept of the virtual enterprise is not new at all. Project industries such as for instance within the construction projects, have been around since for centuries (Vesterager et al, 1999). What is new however is that the virtual enterprise is facilitated by ICT, which enables a higher degree of customization, globalization and outsourcing as described previously.

In this section the virtual enterprise will be described in more detail. First the virtual enterprise is presented in a historical context. This is followed by a list of definitions found in the literature. Based upon these a set of characteristics of what a virtual enterprise is presented along with the definition applied in this research. The section concludes with discussing whether the virtual enterprise is a panacea.

3.2.7.1 Historical context

As indicated above the virtual enterprise is emerging as a new business concept. A lot of attention has been and are given to the virtual enterprises, virtual organizations and the like. In the following a chronological review of the essential authors that have addressed virtual enterprise is presented.

As an actual term the virtual enterprise has only been discussed about a decade. However, scholars have addressed some of the characteristics of the virtual enterprise earlier. In 1984, Miles and Snow introduced a new organization concept as the 21st century firm called *dynamic network*, with the core activating and control mechanisms to be:

> *"Broker-assembled temporary structures with shared information systems as basis for trust and coordination."* (Miles & Snow, 1984)

They later define it as:

> *"a far more flexible structure than any of the previous forms, it can accommodate a vast amount of complexity while maximizing specialized competence, and it provides much more effective use of human resources that otherwise have to be accumulated, allocated, and maintained by a single organization."* (Miles & Snow, 1986)

Although they do not use the term explicitly their dynamic network concept possesses some of the characteristics related to virtual enterprises, such as the broker-assemble, temporary and flexibility.

Davidow and Malone was the first to introduce the term virtual in relation to an enterprise entity. In 1992 introduced the term Virtual Corporation:

> ”*The virtual corporation began as a vision of futurists, became a possibility for business theorists, and is now an economic necessity for corporate executives.*” (Davidow & Malone, 1992)

As previously mentioned, Davidow and Malone claim that *“we are in the midst of a new business revolution”* (Davidow & Malone, 1992). The centerpiece of this business revolution is the virtual enterprise, which is seen as an enabler of the aforementioned *virtual product* (A virtual product is produced instantaneously and customized in response to customer demand). According to them the realization of the virtual product requires changes in the companies:

> *"Building a virtual product will require a company to utterly revise itself, control ever more sophisticated types of information, and master new organizational and production skills."*

Davidow and Malone outline the following characteristics of a virtual enterprise (the following characteristics are extracted from (Davidow & Malone, 1992)): “

- *From the outside as the observer:*
    - *It will appear almost edgeless,*
    - *Have permeable and continuously changing interfaces between company, supplier, and customers*
    - *Appear less as a discrete enterprise and more as an ever-varying cluster of common activities in the midst of a vast fabric of relationships*

- *From inside the firm:*
    - *Constantly reforming according to need*
    - *Job responsibilities will regularly shift*
    - *Even the very definition of employee will change, as some customers and suppliers begin to spend more time in the company than will some of the firm's own workers."* (Davidow & Malone, 1992)

In their 1993 cover story in *Business Week* Byrne, Brandt & Port introduce the virtual corporation. According to them the key attributes of a virtual enterprise are (Byrne et al, 1993):

- Technology (ICT enables companies to link up and work)
- Excellence (each partner brings its core competencies)
- Opportunism (partnerships will be less permanent, less formal, and more opportunistic)
- Trust (these relationships make companies more reliant on each other and require far more trust than ever before)
- No borders (redefines the traditional boundaries of the company)

One of the essential elements of the virtual enterprise is that it is configured from time-to-time based upon a specific customer need (or expected need). Thus the virtual enterprise is a means for organizations to be able to agilely respond to customer needs. Agility and virtual enterprise is thoroughly discussed in the book (Goldman et al, 1995). The days of mass-customization with its "If we can make it, they will buy it" philosophy is being supplanted by the agile competitive environment aiming to enrich the customer (Goldman et al, 1995). It should be noted, that this does not mean that the days of mass-customization are over. What it means is that in more and more situations the customer will require a product customized to his or her specific needs and desires. Compared to other initiatives agility should be treated a strategic initiative addressing the new competitive reality:

"Reforms introduced by companies since the early 1980s to improve their competitiveness – just-in-time, the quality movement, 'lean' manufacturing – have been tactical responses to marketplace pressures. These reforms aim to improve how companies are doing what they are already doing. Although these efforts are appropriate and valuable, they reflect an acceptance of the status quo, rather than a recognition of the need to confront a new competitive reality, one that challenges what companies ought to be doing, not just how they can do a better job of what they are already doing. Agility challenges the prevailing paradigms of organization, management, production, and competitiveness. It is explicitly strategic rather than tactical, taking no established practices for granted." (Goldman et al., 1995)

"The crucial step a company must take, in our view, is to link the evaluation of agility directly to concrete corporate goals and to specific, ***strategic*** *initiatives for accomplishing those goals."* (Goldman et al., 1995)

The authors describe six strategic reasons for use of the virtual organization concept (in quotations): "

1. Sharing infrastructure, R&D, risk and costs
2. Linking complementary core competencies
3. Reducing concept to cash time through sharing
4. Increasing facilities and apparent size
5. Gaining access to markets and sharing market or customer loyalty
6. Migrating from selling products to selling solutions" (Goldman, 1995)

Another author that has addressed the virtual enterprises is Abbe Mowshowitz. In his article in 1994 Mowshowitz tries to identify what the principles that underlie successful firms, which he calls virtual organizations, are. He seeks to build a theory that can account for the phenomenon, i.e.

"Provides a precise definition of virtual organization."

A central element in the definition and the accompanying theory is the distinction between *conception and planning* of an activity and its *implementation* (Mowshowitz, 1994). According to Mowshowitz,

> "A *Virtual Organization is a goal-oriented enterprise (i.e., unit, function, activity) operating under meta-management.*" (Mowshowitz, 2002)

Meta-management is the management of a virtually organized activity. Mowshowitz further defines meta-management as consisting of five basic activities: "

> *(1) Analyzing abstract requirements*
> *(2) Determining the possibilities for satisfying requirements*
> *(3) Tracking allocations of satisfiers to requirements*
> *(4) Maintaining and possible revising the procedure for allocating satisfiers to requirements*
> *(5) Reviewing and adjusting the optimality (or "satisfying") criteria of the allocation procedure.*" (Mowshowitz, 2002)

Thus, Mowshowitz speaks about a set of abstract requirements, which should be fulfilled by a set of concrete satisfiers, which through an allocation procedure is assigned to the abstract requirements. Contributing with their specific core competencies various partners can provide the satisfiers needed to fulfill the requirements. Thus, realization of virtual enterprises requires a shift in the execution of management:

> *"The shift from conventional to virtual organization requires a basic reorientation of management philosophy. Managers must embrace the idea of logically separating task requirements from potential task satisfiers – or of distinguishing between goals and the procedures for implementing them."* (Mowshowitz, 1997)

One of the shifts is that management should establish a mature ability to switch. Preparation is a way to systematize the ability to adapt quickly to new market demands. As addressed previously the new business requirements pressure companies to become more flexible:

> *"Conventional corporate management may occasionally switch between options, but it normally does so on an ad hoc basis, whereas in virtual organization, switching is standard operating procedure; that is, it is a basic management principle of this innovative form of organization."* (Mowshowitz, 1997)

Thus, the ability to respond quickly to new possibilities related to a changing environment becomes the path to success of a virtual enterprise. The aforementioned agility is related to the ability to switch swiftly and the virtual enterprise in general:

> *"Virtual enterprises are seen as a means to realize agile manufacturing"* (Vesterager et al, 1997)

As can be seen from the above description the virtual enterprise is a strategic answer to the previously described business challenges. That it, the virtual enterprise is an aggregation of core competencies from different, partners as a means to create customized (ideally virtual) products to the global market. All this is enabled by technology in general and ICT in particular. The characteristics of virtual enterprises are further elaborated in the following section with a definition of a virtual enterprise used in this research.

3.2.7.2 Virtual enterprise characteristics

As already indicated the virtual enterprise has been devoted a lot of attention in the literature. However, no unified definition has been reached.

> *"The paradigm of virtual enterprise (VE) represents a prominent area of research and technological development for today's progressive industries. So far, there is no unified definition for this paradigm."* (Camarinha-Matos, 1999d)

The fact that no unified definition has been specified can also be seen from the various different (although related) definitions found in the literature. As an attempt to illustrate the different foci in the various definitions the author of this dissertation has underlined different characteristics in each of the following definitions:

- *"It's a temporary network of companies that come together quickly to exploit fast changing opportunities. In a Virtual Corporation, companies can share costs, skills, and access to global markets, with each partner contributing what it's best at." (*Byrne et al, 1993)
- *"The virtual company is a form of joint venture but with the important differences. It is a temporary alliance of member companies which join to take advantages of a market opportunity. The virtual company will pose almost no employees or inventoried resources. Each member company will provide its own core competencies in areas such as marketing, engineering, and manufacturing to the virtual company."* (Barnett et al, 1994)
- *"A virtual enterprise is basically reconfigurable, computer-networked, customer solutions delivery systems."* (Kovac, 1994)
- Virtual Factory *"a community of dozens, if not hundreds, of factories, each focused on what it does best, all linked by an electronic network that would enable them to operate as one – flexible and inexpensively – regardless of their locations."* (Upton & McAfee, 1996)
- *"A virtual organization is a collection of business units in which people and work processes from the business units interact intensively in order to perform work which benefits all."* (Preiss et al, 1996)
- *"A virtual enterprise is to be perceived as a structured and well-considered constellation of parts from a network of 'traditional' enterprises."* (Vesterager et al, 1997)

- *"A virtual organization is a geographically dispersed organization, in which travel for the purpose of exchanging information is minimized."* (Moller, 1997)
- *"A virtual enterprise (VE) is primarily an interoperable network of pre-existing enterprises with a common goal, which can function as a single real organization"* (Afsarmanesh et al, 1997)
- *"A Virtual Enterprise is a temporary organization of companies that come together to share costs and skills to address business opportunities that they could not undertake individually."* (NIIIP, 1998)
- *"The virtual enterprise is based on the ability to create temporary co-operations and to realize the value of a short business opportunity that the partners cannot (or can, but only to lesser extent) capture on their own."* (Katzy & Schuh, 1998)
- *"A virtual enterprise is a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks."* (Camarinha-Matos, 1999d)
- *"A VE is a temporary aggregation of core competencies and associated resources collaborating to address a specific situation, presumed to be a business opportunity."* (Goranson, 1999)
- *"A business created largely out of purchased products, components, and services."* (Porter, 2001)
- *"Geographically distributed, functionally and culturally diverse, dynamic and agile organizational entities linked through ICT."* (VOSTER, 2001)

As can be seen from the above definitions although alike various scholars emphasize different aspects of a virtual enterprise. Based upon the literature review the general characteristics of a virtual enterprise are identified to be:

- Temporary nature (set up with a clear purpose)
- Aggregation of partners (not necessarily know each other previously, focus on core competencies)

- Quick respond to a business opportunity (customization, from product towards solutions, reconfigurable and agility)
- Partners have a common goal (opportunity the partners could not undertake individually)
- Sharing of cost and skills/competencies (need for cooperation, competence unfolding)
- Supported/enabled by ICT (enabling virtuality and globalization)
- Trust (prerequisite for a successful partnership)

In this dissertation the following short definition of a virtual enterprise will be applied:

> *"A Virtual Enterprise is a customer solution delivery system created by a temporary and re-configurable ICT enabled aggregation of core competencies."* (Sari, 2004)

3.2.7.2.1 Panacea

Based upon the descriptions in the previous section one could get the impression that the virtual enterprise could be considered as a panacea solving all problems related to the new business challenges described previously. This however is a modification of the truth.

Not surprisingly, there are challenges related to organizing the business as a virtual enterprise. One of the prerequisites for participating in a virtual enterprise is that the company can master partners. The explosive expansion of ICT through the 1990s generated a lot of fuzz about the potential and the wonders of the emerging virtual world. Up to the burst of the 'IT-bubble' acronyms such as Business-to-Customer and Business-to-Business were devoted a lot of attention.

As a respond some authors such as e.g. Chesbrough & Teece claim that there was a tendency to over-emphasize the wonders of the new emerging virtual world:

“Virtues of being virtual have been oversold. Those rushing to form alliances instead of nurturing and guarding their own capabilities may be risking their future" (Chesbrough & Teece, 1996)

As shown on Figure 3.2 the decentralized virtual company has the highest incentives to take risks. However the virtual company also possesses the lowest ability to settle conflicts and coordinate activities due to its decentralized nature. The integrated corporation is at the other end of the extreme as showed on the figure.

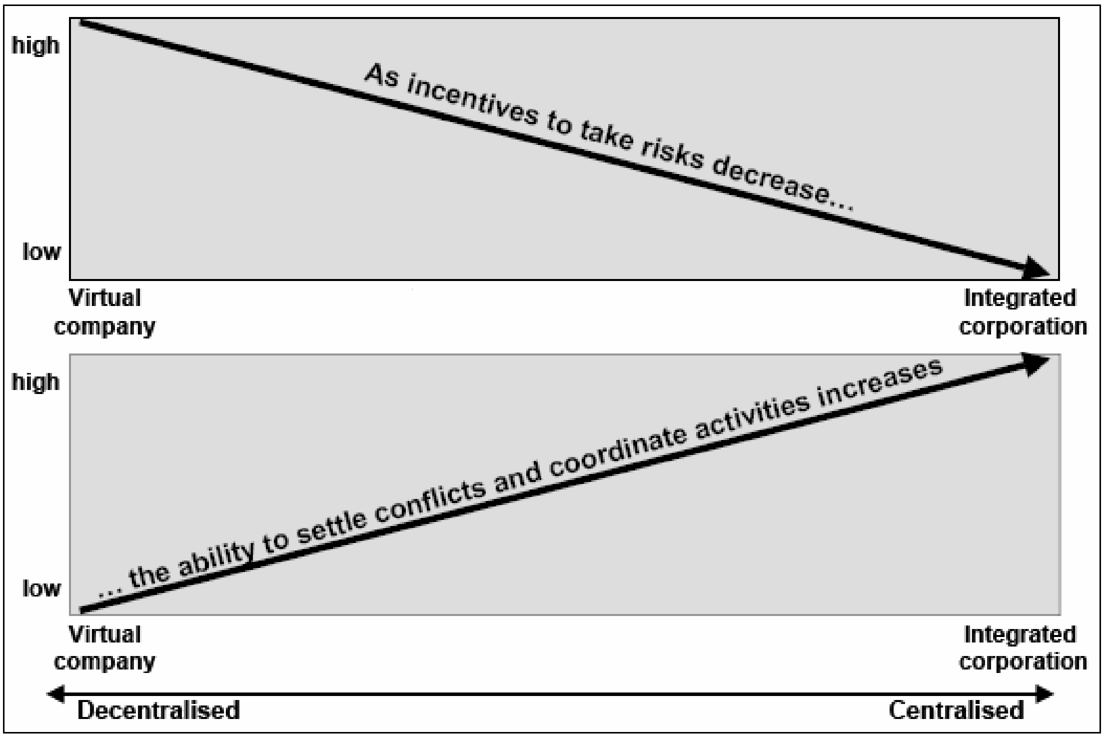


Figure 3.2 Finding right degree of centralization

The authors claim that the virtual approach to innovation poses serious strategic hazards if applied for systemic innovations due to their requirements for information and coordinated adjustments throughout their entire development.

"The virtual companies that have demonstrated staying power are all at the center of a network that they use to leverage their own capabilities. … the

virtuous virtuals have carefully nurtured and guarded the internal capabilities that provide the essential underpinnings of competitive advantage. And they invest considerable resources to maintain and extend their core competencies internally" (Chesbrough & Teece, 1996)

Another related hurdle to virtual enterprises is the trust issue related to cooperating with new and unknown or less familiar partners. Inclusion of new partners possessing the competencies required in a specific situation is seen by many as one of the major challenges related to virtual enterprises.

When presenting the virtual enterprise concept, as described above, to industry some of them tend to assess it as academic thoughts not reflecting their real life situations. They claim that it usually takes years to establish a trusted relationship with a new partner. This is especially the case if the partner is to be considered as a strategic partner, with whom e.g. shared development work can be carried out, and with which a tighter integration can be established, e.g. shared ICT infrastructure, procedures, rules, etc.

3.2.7.3 Categories of virtual enterprises

Although there is no strict academic definition regarding VE, different VE models feature common business and technical characteristics and attributes. Deploying the above specified criteria as classification criteria, two well-defined categories of VE can be identified (Ouzounis & Tschammer, 2000; Malone & Crowston, 1991; Banahan & Banti, 1999; Stricker et al, 2000). These are:

- Static Virtual Enterprises,
- Dynamic Virtual Enterprises.

In the following section, these two generic VE categories are further analyzed and certain examples are provided to better clarify the different models, concepts, and benefits and drawbacks that both approaches share.

3.2.7.3.1 Static virtual enterprises

In Static Virtual Enterprises (SVE) a set of business partners is linked together in a static and fixed way, i.e. the shared business processes are tightly integrated. The business relationships among the partners, i.e. the process interfaces are pre-defined, tightly coupled, fixed, wellintegrated, and customised among the partners (NIIP, 1996; Malone & Crowston, 1991). The network is fixed and pre-determined and thus, the structure of the VE is static and predetermined. Based on the distribution and management style of the network, two types of Static VEs can be identified, namely centralised and decentralised (Ouzounis & Tschammer, 1998a; Stricker et al, 2000).

In the Centralised Static VEs (CSVEs) a dominant business domain co-ordinates the business relationships among the members of the network, enforces the technical interfaces for business integration among the partners, integrates the processes of the partners by creating shared processes, and manages the underlying technical infrastructure and the shared business processes of the partners in a static and centralised way (NIIP, 1996; Wognum et al, 1999a & b). Partners and the central organisation form long-term relationships and focus on investment returns over the lifetime of that relationship. Finally, the establishment of the VE is performed manually, in a customised way, and under the full control of the dominant organisation. The required integration, development, and re-engineering costs are high for all members (McCaffer & Garas, 1999; Nwana et al, 1999).

Typical examples for CSVEs are models that have been applied in the automotive manufacturing business (Zarli & Poyet, 1999; Geppert, 1998). In that case, a big automotive manufacturer has a network of suppliers, distributors, and re-sellers that are working together in different phases of the production, distribution, and reselling process. The big manufacturer has specific needs and requirements and enforces his requirements in order to increase the degree of automation and decrease the production and distribution costs. The network of suppliers, resellers, and distributors closely co-operate with the central dominant business domain by adopting and integrating the pre-specified interfaces.

In Decentralised Static Virtual Enterprises (DSVE) different business partners are linked together in a rather autonomous and decentralised way. This type of network is similar to the previous one except that there is no central, dominant, management organisation and each member of the network may co-operate with many other domains (Malone & Crowston, 1991; Ouzounis, 1998c).

None of the partners has full control over the network and the underlying infrastructure, while integration among the business processes of the members is performed in a jointly, coordinated, and incremental way. Partners form long-term business relationships and gain investment returns over the lifetime of those relationships (Mohan & Krithi, 1998; Fredederix, 1998). Finally, the establishment of the VE is performed manually and in a customised way addressing the specific technical requirements of the partners. The development and integration costs are rather high, while the evolution of the network is rather impossible (Zarli & Poyet, 1999; Wognum et al, 1999a).

High tech manufacturing today exemplifies this model. Organisations such as semiconductor fabs and board assembly houses focus on one activity in a complete value chain and then partner with multiple other organisations in order to play a role in multiple value chains. Every partner plays a role in the VE and contributes primarily its own core competencies, i.e. business processes and resources. In high Tech manufacturing the VE members can work on the production and assembly of new products, as well as, on the distribution of products to different re-sellers.

A more recent approach to automate the process of forming a static VE is to use a virtual marketplace or a directory service where potential VE members register their resources and business processes (CrossFlow, 1999; Tombros, 1999; Tombros & Geppert, 2000; Ouzounis, 1998b). The virtual marketplace provides matchmaking services to business domains that want to locate VE partners (Spinosa et al, 1998;

McCaffer & Garas, 1999). Human operators searching the marketplace and locate potential partners that can provide specific processes. Then, a manual, human-driven negotiation process starts for the selection of the most appropriate VE candidate partner. With this approach, the time required to find partners and establish business relationships is improved. This approach takes advantage of the new, open, Internet economy and significantly improves the formation process of the virtual organisations. However, after the formation of a VE, the business relationships among the partners, i.e. the interfaces among the shared business processes, remain static and fixed, while the evolution of the VE in terms of new members, that might provide better processes with better terms, is rather impossible (Hoffner & Schade, 1997; Hoffner & Crawford, 1998; Hoffner, 1999).

Marketplaces can be used in a more effective and dynamic way not only in the formation phase of VEs but also in the execution phase (Ouzounis & Tschammer, 1999b). This means that the partners that are involved in the provision of the shared process are changing continuously and dynamically according to the requirements of the customer and the processes. In that case, for every business process execution a new VE is being created in a dynamic way addressing the needs and requirements of the customer and the individual partners. The deployment of marketplaces not only for the establishment of VEs but also during the provision of shared processes can lead to significant improvements (Ouzounis & Tschammer, 1999b).

3.2.7.3.2 Dynamic virtual enterprises

In Dynamic Virtual Enterprises (DVE) a set of business partners is linked dynamically, on demand, and according to the requirements of the customers, by deploying a virtual marketplace.

The business domains do not have fixed business relationships and thus the VE is not static and might change continuously based on market driven criteria (Ouzounis & Tschammer, 1999b; Fielding, 1998; Doz & Hamel, 2004).

The virtual marketplace provides services for the registration of partner process offerings based on some generic, well-known, globally specified process templates. Business domains that want to form VE relationships can register offers on the marketplace related to certain process templates. Whenever a business domain wants to use a particular process, searches the marketplace, and locates all the potential partners that can provide the service. As soon as the list of VE candidate partners for one particular process has been found, the selection process starts. The selection process between the domains is usually performed through negotiation. The negotiation process might be either, manual, or automated, while the result of it is usually a short term contract that regulates the business relationship among the involved domains (Geppert et al, 1998; Grefen et al, 1999; Weitzel et al, 1999).

By deploying virtual marketplaces, there are no explicit static business relationships among the partners and thus, no integration among the processes of the partners is required. Marketplaces are usually organized around certain globally specified service or product templates that can be offered by the different vendors. The marketplace is a match making mechanism that brings potential process providers together with potential users of these processes. Although marketplaces and matchmaking mechanisms have been used for some time for business to consumer electronic commerce purposes (Ebay, 1998) they have not been actually deployed for dynamic VE purposes (Ouzounis & Tschammer, 1998b). The main reason was the lack of technologies that enable the easy and flexible definitions of process templates, mechanisms for automated negotiation, and autonomous interaction among different domains. Due to the advent of eXtensible Markup Language (XML) (W3C) and its ultimate acceptance, as Internet metalanguage, concepts like virtual marketplaces have started to appear (Ouzounis, 2000; Zarli & Poyet, 1999; Mitrovic et al, 2003).

The primary focus on virtual marketplaces is on efficiency of transactions and maximization of value per cost of each vendor's offer. Organizations may participate in the marketplace only briefly or they may be long term members. Relationships between process users and process providers tend to be short term. Thus, investment returns are gained over single transactions, as well as, over the time span of the marketplace participation. The number of members of the network can easily change and thus, the structure of the VE can change from one service provision to another according to the specifics of the customers and the current needs of the members. This is a significant evolution mechanism that takes advantage the demand and supply, i.e. the process offerings by the individual domains.

Based on the distribution and management style of the network, two types of Dynamic VEs can be identified (Malone, 1991; Alonso, 1998):

- Centralized, when the owner of the marketplace is a VE partner. This domain manages and administers the virtual marketplace and enforces specific process templates. Although, from technical point of view, it is possible to organize a VE in terms of a Centralized Dynamic VEs (CDVEs), from business point of view is rather unusual. The main reason is that the marketplace should be a trusted, third party provider that is not involved into the VEs. Centralized dynamic VEs can be deployed by very big organizations that would like to transit from the Centralized Static VEs into more dynamic cases (Zarli & Poyet, 1999; Geppert et al, 1998).
- Decentralized, when the owner of the marketplace is a third party provider that has no relationship to the registered partners. This is probably the most advanced and flexible model that features the most benefits. However, the required business systems and technologies are far too complex and for the moment immature (Ouzounis, 1998c).

An interesting area where Decentralized Dynamic VE (DDVEs) concepts are applied is the area of trading communities. A characteristic example is the area of logistic companies. In that case, logistic companies can register their processes into a specialized marketplace. A potential process might be the delivery of parcel where the properties of the process might be the reached destinations, the price, the time needed to transfer the parcel, the offered guaranty, the transformation media, etc. Then, business domains that want to use a logistic service, search automatically on the trading community, select the best partner that exist at this moment on the marketplace, based on certain requirements, and use the service. For the initial customer of the VE the whole process is totally transparent. As the offers in the marketplace change, i.e. new companies register and deregister with better terms, conditions and prices, then the selection of the best partner depends on negotiation practices. The VE might exist for only one service invocation or for more. However, if companies want to take advantage of the market conditions enabled by Internet-based commerce, they should frequently deploy the capabilities of the virtual marketplace in order to get better prices and quality of service.

The above scenario illustrates the key elements of DDVE. The VE usually exists only for the duration of a single service provision. Certain selection and negotiation requirements specify the VE partner that will be selected each time (Billington, 2004). The evolution of the VE is granted due to the loosely coupled relationships among the partners and the marketplace capabilities. The registration of process offerings on the marketplace is based on globally specified service templates (Tombros & Geppert, 2000; Wognum & Faber 1999b). The marketplaces are becoming more specialized and closely related to specific industrial sectors. In long term, special trading communities for specific industry sectors will be created. The form and relationships among the partners of the VE can change continuously. The process offerings registered in the marketplace can change dynamically and on-demand according to the demand and supply (Ouzounis, 2000b).

It is obvious that dynamic VEs improve significantly the static ones and take full advantage of the open, global, opportunities offered by the Internet and the global economy. In the following section a more formal and focused assessment of the static and dynamic VEs is provided.

3.2.7.3.3 Evaluation of virtual enterprise categories

The classification and assessment of the basic VE models, proposed so far, will be done with the previous mentioned classification criteria (Camarinha-Matos & Afsarmanesh, 1999c; Malone & Rockart, 1991; Mitrovic et al, 2003). Therefore, a comparison of each model against the classification criteria leads to the following conclusions:

- **Lifetime of the relationship:** DVEs feature very short lifetimes, while SVE feature longer ones. In SVE case, the relationships are static, well-integrated and thus, no flexible enough for alterations, modifications, and evolution. This dimension also determines the time period over which investment returns must be achieved,
- **Degree of integration**: Tightly coupled SVEs, which function essentially as a single virtual organization, exhibit high process integration between partners. Loosely coupled DVES are at the far end of the spectrum and show very low process integration between the partners,
- **Number of VE Partners:** In SVEs, the number of partners participating in the VE is static and pre-determined due to the specialized integration activities required. In DVEs the number can change dynamically, upon demand and supply, and based on the requirements of the individual members of the marketplace,
- **Degree of Autonomy:** SVEs require high degree of integration among the partners and thus, the degree of autonomy is rather low. The business processes of one partner are highly depending on the others. On the contrary, DVEs feature more autonomicity because the relationships among the partners are not static and well-integrated. Thus, any changes to business processes can easily be done.
- **Degree of Distribution**: All the above mentioned models have a good level of distribution among the business processes of the partners. However, SVEs are based on a centralized dominant model, while DVEs reveal, due to the nature of the model, the highest level of distribution and autonomicity among the business processes and partners.
- **Degree of Evolution and Scalability:** In SVEs the relationships among the partners are static and thus the level of scalability is low. It requires high development costs to re-design the network and change the interfaces among the partners. On the contrary, on DVEs there are no tightly coupled interfaces among the partners and thus, scalability and business evolution is a key issue.

- **Focus on process efficiency vs. focus on per-transaction efficiency and value:** As we saw above, partners that work as part of a larger virtual organization focus on achieving overall process efficiency. Partners that work on a per-transaction basis need to focus on achieving efficiency and value within the individual transactions.

In the following Table 3.1, a summary of the above analysis and discussion is illustrated.

Table 3.1 Comparison of VE Categories

	Dynamic Virtual Enterprises	**Static Virtual Enterprises**
Lifetime	Low	High
Integration	Low	High
Number of Partners	Dynamic	Static
Autonomy	High	Low
Distribution	High	Medium
Evolution/Scalability	High	Low
Process Efficiency	Medium	High
Transaction Efficiency	High	Low

Based on the above selection and categorization criteria, it is obvious that DVEs are a more promising business model with a lot of interesting features. Due to the open mechanisms of Internet economy dynamic, flexible, autonomous VEs that take advantage of the market conditions are preferred.

Although from business point of view DVEs are the most promising business model, from technical point of view the required technical solutions and systems are more complex, sophisticated and distributed (Camarinha-Matos & Afsarmanesh, 1999b; Alzaga & Martin, 2004). However, the advent of Internet and open communication protocols, like TCP/IP and HTTP, distributed middleware systems, like CORBA-IIOP and Java RMI, and extensible meta languages, like XML, provide the basic building blocks for the development of management platforms that will realize the concept of DVEs.

3.2.7.4 Infrastructures & interoperability

3.2.7.4.1 Trends in software development for VE infrastructures

Although the potential advantages of the Virtual Organizations are well known at the conceptual level, their practical implantation is still far from the expectations. In order to leverage the potential benefits of the agile VO/VE paradigm, there is a need for flexible and generic **ICT infrastructures** to support the full life cycle of the VO/VE and the establishment of effective **interoperability** mechanisms.

Some of the main trends in terms of **software development** towards generic VO/VE infrastructures include:

- *Layer-based (transactional) frameworks* - which add layer to the existing ICT platforms of the enterprises. Inter-enterprise co-operation is then performed via the interaction (transaction-oriented) through these layers. These frameworks include client-server solutions, web-enabled servers, and frequently adopt workflow models and standards for information exchange.
- *Agent-based frameworks*, including those approaches, which represent enterprises as agents and the inter-enterprise co-operation as interactions in a distributed multi-agent system.
- *Service-federation / service-market frameworks* - according to which enterprises publish their services in service directories, representing their potential "offer" to the cooperation processes. By means of proper "standard" service interface, the interoperability with other (remote service requesting) enterprises, regardless of the heterogeneity associated with the actual implementation of the services themselves, is supported.

These frameworks are not necessarily exclusive and thus mixed approaches can be found. Different organizational forms and different management policies can, in principle, be supported by the same basic infrastructure.

Table 3.2 Transaction-oriented layer-based horizontal infrastructure

Transaction-oriented layer-based horizontal infrastructure	
Key facets: ❑ Safe communications ❑ Cryptography, symmetric & asymmetric keys, digital signature, certificates ❑ VPN ❑ Information sharing and exchange ❑ Distributed/federated information management ❑ Specification of access rights / visibility ❑ Workflow-based coordination ❑ Standards for exchange of some classes of information ❑ EDIFACT, STEP ❑ More recently XML based structures ❑ Various approaches for remote objects & services access ❑ RPC, CORBA, RMI, EJB, Jini	**Current limitations**: ❑No common reference model ❑ Need to integrate different technologies (from different vendors) ❑ Technical complexity ❑ Unclear responsibilities ❑ Infrastructure is still complex, difficult to configure and poor interoperability ❑ Limited support for distributed business process management ❑ Lack of support for VE dissolution ❑ Limited mechanisms for tracking and auditing ❑ Poor support for breeding environments management

Table 3.2, 3.3 and 3.4 summarize the main facets addressed by VE projects and also the current limitations of the proposed infrastructure solutions (Camarinha-Matos & Afsarmanesh, 2003).

Table 3.3 Agent-based horizontal infrastructure

Agent-based horizontal infrastructure	
Key facets: ❑Support for VE creation ❑ Partner search and selection based on negotiation ❑ Virtual market places and brokers ❑ Preliminary steps towards e-contracting ❑ Some support for VE operation ❑ Dynamic scheduling functions ❑ Combination of inter-agent communication and federated information management ❑ First steps in contract management	**Current limitations**: ❑There are many development platforms for MAS, namely some FIPA compliant (e.g. JADE, FIPA OS) but they are not robust enough when operating over Internet ❑ Security and persistence mechanisms are not yet well integrated with MAS ❑ Lack of integration between AI and BP communities (e.g. there is a need to integrate ACL with BP languages) ❑ Developments mostly at prototype level; real demonstration cases missing

Table 3.4 Service federation horizontal infrastructure

Service federation horizontal infrastructure	
Key facets: ❑ Basic architectures for service federation ❑ Mechanisms for remote service invocation ❑ Preliminary standards for service description and cataloguing: UDDI, WSDL, SOAP, … ❑ Basic service search mechanisms ❑ Preliminary mechanisms for Value Added Service composition	**Current limitations**: ❑ Poor integration of service federation and VO concept ❑ Poor integration of security / privacy mechanisms ❑ Search mechanisms still too basic ❑ Ad-hoc concept of portal

3.2.7.4.2 Infrastructures for professional virtual communities

When a proper cooperation environment is in place, professional **virtual communities** (PVC) / virtual teams emerge within such environment, constituting a fundamental element of value creation, innovation and sustainability. A large number of computer supported cooperative tools are becoming widely available for synchronous cooperation. Considering the geographical distribution, the autonomy of the VE members, the local corporate cultures, and also the individual working preferences of the team members, it is likely that most of the activities will be carried out in an asynchronous way, which requires new assisting tools. In terms of coordination, several approaches to develop *flexible workflow* systems have been proposed. In the case of processes mainly executed by humans, rigid forms of procedural control are not adequate. People like to keep their freedom regarding the way they work. Product design, like any other creative process evolves according to a kind of “arbitrary” workflow. It is therefore necessary to also support *loosely constrained* sets of *business processes*. Table 3.5 summarizes the main facets addressed by VO projects and also the current limitations of the proposed solutions in the support for professional virtual communities.

Table 3.5 VCP support infrastructure (Matos, 2003)

VCP-support infrastructure	
Key facets:	**Current limitations**:
❑First steps towards "shared working spaces"	❑Very limited integration of tools and mechanisms
❑ Large number of "small" tools (e.g. chat, instant messaging, teleconference, CSCW)	❑ Limited coordination facilities
❑ Some application sharing mechanisms	❑ No adequate VCP management for professional communities (Virtual Communities of Practice - VCP) able to capture multi-level relationships
❑ First steps towards flexible workflow and asynchronous coordination of activities, notification mechanisms	❑ No integration of IPR issues in the VCP management services
❑ Basic VCP management	❑ Limited support for mobile contexts

Advanced forms of cooperation mostly in the area of design and manufacturing require mechanisms to support a controlled "intrusion" of a company, for instance the VO coordinator, into the "territory" of its partners. Consider the case that a company wishes to "open a window" over the shop-floor of its partner (according to contractual rules) to monitor the manufacturing process of the ordered parts and even have an interference on the shop floor processes, i.e. supervise these processes from distance and in cooperation with the local people. The local supervision functionalities installed in each production site shall interact / cooperate with the global VO (network-wide) supervision, providing controlled levels of transparency.

3.2.7.4.3 Conclusions on infrastructures & interoperability

A large number of research initiatives and industry cases have been developed worldwide during the last years. VOSTER analysis included about 35 projects developing collaboration models and supply chain management systems (Collaboration models EXTERNAL, ECOLNET, E-COLLEG, DYCONET, WHALES, SCOOP, LENSIS, UCANET, LINK3D, SCM, ADRENALIN, AITImplant, APM, BURMA-X, CASH, CENNET,CHAINFEED, COCONET , COMPANION, CO-OPERATE, CUTTING-EDGE, DEEPSIA, DILEMMA, District, DOMINO, DYCONET, ECSR, EDIBOLD-SCS, EFLORA, EUROSHOE, EXTERNAL ,GEM-EUROPE,INTELISHOE, LIAISE , LOGSME, MEDIAT-SME, OBELIX, PABADIS, ProdChain, SCOOP, SMARTISAN, UCANET, V-CHAIN, XML-KM … Project for industry sector applications: EUROSHOE, CARUSO, E-CONSTRUCT, E-FLORA, GLOBEMEN, ISTFORCE, PABADIS, SUMMIT, WHALS).

Based on the review of these cases some critical elements in establishing ICT infrastructures were identified. The EU and non-EU feedback supported the VOSTER observations:

1. The implementation and configuration of a VE *support infrastructure* still requires considerable engineering effort, an obstacle to dynamic VEs. Even the most advanced infrastructures coming out of leading R&D projects require *complex configuration* and *customization* processes hardly manageable by SMEs.

Almost all VO/VE projects have been forced to create their own infrastructure due to the lack of common reference architectures and interoperability principles. Only a few projects if any, have addressed the development of a comprehensive horizontal infrastructure. There is a need to discern between enabling vs. disabling technologies; some efforts are too biased by short-term technologies, which might represent an obstacle for non-ICT SMEs.

Interoperability still remains as a central question in systems integration and, the fast evolution of ICT technologies with reduced life cycles emphasis the need to cope with technologies with different life cycles and at different stages of the corresponding life cycle.

Emerging infrastructures induce new organizational forms, but emerging organizational forms also require new support infrastructures.

2. In addition to the infrastructure needs, some commonalties on basic functions start to emerge, but there is a lack of general *support functions* for collaborative distributed business processes. It is still the case that some stages of the VE/VO life cycle are mostly ignored by current projects. Efforts have mainly been focused on the "Operation phase". Some recent projects also emphasize the "Creation phase" with some impact also on the "VE/VO Evolution", but the "Dissolution phase" is almost not touched.

3. *Application integration* is concerned with the usage of ICT to provide interoperation between enterprise resources. Cooperation between humans, machines and software programs has to be established by the supply of information through inter- and intra-system communication. Application integration is split in two parts. Whereas semantic standards support integration at the level of 'meaning', syntactic standards are meant for integration at the level of 'form'. Syntactic standards enable sources and messages to have similar formats. Standards in this area are STEP Part 21, Java RMI, XML, Corba, and DCOM. Semantic application integration should result in a situation where the output of applications is meaningful to other applications. Examples of standards at this level are EDIFACT, STEP Application Protocols, RosettaNet, and BizTalk.

3.2.7.5 Objectives of SMEs in VE

Several factors closely related to the size of SMEs restrict their competitive position (Kerste & Muizer, 2005). Their limited capacity may prevent them from taking on large tasks and to provide large volumes of standardized products at low costs, which limits their capacity to exploit market opportunities. Or they may not have all the specialized competences that are necessary to take on new tasks, which limit their capacity to adapt to changing market conditions. In these respects SMEs have competitive disadvantages compared to LSEs regarding advantages of scale as well as of scope. However, SMEs have competitive advantages over LSEs when it comes to flexibility and responding to needs in small market niches (Storey, 2003).

One answer to overcome the effects of limited resources is to co-operate with other enterprises. Through cooperation SMEs may gain control over external resources that larger enterprises control through their hierarchy (Wallenklint, 2001). Thus SMEs may achieve similar conditions for their value creation as large companies.

Co-operation will usually be understood as a relationship between independent enterprises or partners that combine their efforts and resources in a value creation process. As such, co-operation has a specific purpose although this purpose may be different for the partners. Co-operation is the link between the partners. The structure consisting of the partners and their co-operation is a partnership.

The objectives of VE or SME co-operation are intrinsic to each enterprise. Each partner may have different objectives. One enterprise may see the co-operation as a reliable instrument for developing and supplying vital components. The other partner may look upon the co-operation as a means of securing long-term niche markets. This also illustrates that co-operation is not an objective in itself, but a tool or instrument to reach other objectives. The literature on VE discusses a number of objectives, which can be grouped in four categories:

- Need to secure resources, e.g. labor and capital;
- Reduced transaction costs;
- Efficient access to markets;
- Learning and access to technology.

The objectives listed above are to a large extent interrelated and the distinction between them may be a matter of choice of perspective. E.g. 'efficient access to markets' may follow as a result of better knowledge gained through 'learning' and will in the long run lead to 'reduced transaction costs'. The discussion below summarizes some of the main approaches represented in current research on the topic.

SMEs are, as all enterprises, constrained by *scarce resources.* The smaller they are, the more these scarce resources limit the scope as well as the volume of the tasks they can successfully perform. The need to secure resources is often given as a main motive for SMEs to cooperate in a VE platform (Isaksen, 2002; Ahokangas, 2004), but this is also contested and some authors suggest that the resource dimension is overemphasized (Wallenklint, 2001). External networks, based on inter-organizational activities, are therefore often seen as an alternative organization model and have been suggested to enable an entrepreneurial growth strategy (Lechner & Dowling, 2003). Access to various types of resources is therefore an essential issue in the ENSR Enterprise Survey 2003.

The reduction of transaction costs through networking has been a central research topic of the Uppsala School. Transaction costs are the costs involved in establishing a transaction: ex ante costs to search for the product/service, and to establish the transaction; the costs of the transaction itself (contract/agreement), as well as the costs involved in monitoring and enforcing the contract. Indirect costs related to risk (associated with trading with unknown partners and purchasing unknown products) and the costs made to minimize risk are also part of the transaction costs (ENSR, 2003).

Purchasing products and/or services from a co-operation partner may not be at the lowest market price. Cooperation is built on more than only the lowest price. A higher price may be acceptable if co-operation would induce other benefits. Partnerships will e.g. extend the information channels and increase the competence of an enterprise. This will improve the quality and/or quantity of the information available, and the partners will share part of the costs incurred. Furthermore, from a transaction cost perspective, a partnership will allow the enterprise to substitute detailed knowledge of the outcome of a transaction with knowledge of the partner's capability and trustworthiness. This reduces risk as well as administrative costs of the transaction. In a co-operation governed by trust, the enterprise manager can obtain a desired product or service from another enterprise without fearing opportunism (Hakansson & Snehota, 2005). In other words, if the purchaser knows from previous experience that the supplier of a product can be trusted, there is less need for control of the delivery than if the supplier has no track record. Long-lasting co-operations provide this experience and add the element of interdependence and personal relationship.

Cost reduction as a result of co-operation can be observed empirically as reduced costs for transport, production, marketing, technology development etc. Still, most of the benefits are indirect and hard to measure in economic terms at enterprise level. These are the benefits stemming from reduced risk or from access to information.

The co-operation between two Norwegian manufacturers is an example where cost reduction is quite obvious (Welter & Havnes, 2000). The main product of enterprise A needed replacement, but the enterprise had insufficient resources to develop and meet certification requirements for a new product. Enterprise B had a product that fitted with its market profile, but not in its manufacturing facilities. The two enterprises entered into a co-operation where enterprise A manufactures the product, which is marketed and sold by enterprise B under proprietary label. The agreement includes incentives for A to improve the product based on previous experience. The two enterprises clearly understand that they now depend on each other, that this co-operation is mutually profitable and that constant development with input from both partners is required to sustain the profits from this co-operation.

Involvement in SME co-operation is, in other words, a strategic choice to improve the current competitive position of an SME. Equally important is that through better and more efficient access to resources and markets, partnerships will sustain the long-term development and growth of the enterprise. Co-operations are an avenue for learning and internalization of new skills (Morgan, 2002). Well-chosen partnerships make it possible to bypass slow and costly efforts to build one's own capabilities and access to new opportunities.

In the network perspective collective learning and network capabilities refer to what the collective of enterprises - the network - knows about production of goods and services, the organization of production (network capabilities), and how they in consort learn about it (collective learning). Lechner defines network capabilities as activities that are related to access to efficient factor markets at relatively low transport costs, to benefit from migration of engineers among enterprises, to be able to tap into a pool of skilled labor, to share in standardization, to benefit from the general presence of trusting relations etc (Lechner & Dowling, 2003).

An example of such co-operation is enterprises suffering from a limited supply of skilled labor, that decide to set up joint schools and training facilities to qualify workers, rather than to compete on the labor market with each other. Table 3.6 illustrates the relationship between the size of the enterprises and their rationale for SME co-operation. The main features have many similarities for the three size categories of enterprises. Access to new and larger markets is among the three most frequent reasons in all size categories, and access to capital is the least frequent reason in all categories.

Table 3.6 Reasons for SMEs' cooperation by enterprise size (ENSR, 2003)

	Micro enterprises 0-9 employees	Small enterprises 10-49 employees	Medium-sized enterprises 50-249 employees
1st ranked reason	Access to new and larger markets	Access to new and larger markets	Reduced costs
2nd ranked reason	Broader supply of products	Additional production capacity	Access to know-how and technology
3rd ranked reason	Access to know-how and technology	Reduced costs	Access to new and larger markets
Last ranked reason	Access to capital	Access to capital	Access to capital

* Multiple answers allowed.

Source: ENSR Enterprise Survey 2003. Only co-operating enterprises.

In addition to access new and larger markets micro enterprises tend to look at co-operation as a way to extend their scope of products, technology and know-how. Small enterprises often have more focus on production capacity and cost aspects. Medium-sized enterprises focus most frequently on cost reduction and then access to know-how and technology.

The reasons for SME co-operation also differ between industries. For the manufacturing industry, additional production capacity was almost just as important as access to new and larger markets. For the construction sector, access to labor and additional production capacity are supplementary reasons to co-operate with other SMEs. The wholesale and retail sector have a similar pattern, a broader supply of products and access to new and larger markets are again important, but the reduction of costs is another important reason for co-operation. The business and personal services sector listed access to know-how and technology among their top reasons to cooperate.

3.2.7.6 Obstacles to Co-operation

The literature on VE shows that the research so far has mostly focused on the rationale for and the factors that are conducive to co-operation and not on the barriers to co-operation. The obstacles to co-operate will influence the extent of VE. The ENSR Enterprise Survey 2003 examines the perceived barriers to SME co-operation. Table 3.7 illustrates how the perceived barriers are influenced by firm's size. There is a striking similarity over all size categories in their wish to maintain independence. The drive for independence appears to be a strong motivating element in SMEs, which could also explain why most SMEs continue as independent entities instead of merging to achieve benefits of scale. In this perspective VE represents an alternative where some scale advantages can be realized while maintaining a high degree of independence. The only difference between the three size categorize is that the micro firms are less frequently worried about disclosing sensitive information. Managers of micro firms also more often say that they find no barrier to SME co-operation.

Table 3.7 Important barriers to cooperation, by enterprise size (ENSR, 2003)

	Micro enterprises 0-9 employees	Small enterprises 10-49 employees	Medium enterprises 50-249 employees
1st ranked barrier	Wish to maintain independence	Wish to maintain independence	Wish to maintain independence
2nd ranked barrier	Lack of information on suitable SMEs with whom to co-operate	Do not wish to disclose sensitive information to other SMEs	Do not wish to disclose sensitive information to other SMEs
3rd ranked barrier	Do not wish to disclose sensitive information to other SMEs	Lack of information on suitable SMEs with whom to co-operate	Lack of information on suitable SMEs with whom to co-operate

Source: ENSR Enterprise Survey 2003. All enterprises.

Industry sector has a small influence on the perceived barriers for engaging in co-operation with other SMEs. Enterprises in manufacturing and construction sector have a stronger wish to maintain independence than their colleagues in other sectors. It is also the enterprises in manufacturing and construction that perceive more barriers to co-operation and that are most afraid of disclosing sensitive information.

In order to facilitate the interpretation of the barriers perceived by SMEs, two groups of barriers have been identified by factor analysis:

1. Fear to loose independence (including: wish to maintain independence, do not wish to disclose sensitive information, high risk involved): Whenever there is a choice, SMEs seek to maintain internal control of core competences essential to their competitive strength. Co-operation with partners will preferably be used to supplement or improve the internally controlled core competences (Wallenklint, 2001).
2. Institutional barriers (including: lack of information on suitable partners, language and cultural barriers, restrictions imposed by taxation or legal restrictions). The *institutional barriers* reflect common perception among the enterprises on structural aspects regarding the framework for co-operation. The enterprises that mention lack of information on suitable partners also tend to mention language and cultural factors as well as taxation and legal restrictions as barriers.

3.2.7.7 State of art in VE

A rapidly increasing number of projects and R&D activities worldwide are addressing different technical and business aspects of virtual enterprise technologies and infrastructure.

The National Industrial Information Infrastructure Protocols (NIIP) project started at late 1995 in the USA and it was perhaps the first biggest and most significant project in the area of VE. In reality, NIIP is more a work program than one consistent project. NIIP intends to support the formation of industrial VEs and to provide technologies that allow VE participants to collaborate within a heterogeneous computing environment. In its general scope, NIIP addresses the complete VE life-cycle, i.e. establishment, execution, and completion. The NIIP bases its developments on open, standard, core technologies such as the Internet and CORBA, related distributed object oriented technologies, product modeling and description techniques, like the Exchange of Product Model Data standard, and information modeling technologies, like workflow management systems (Georgakopoulos, 1998). Based on this reference architecture, a number of pilot projects have been launched to develop prototypes, e.g. SMART, Solutions for SME Adaptable Replicable Technology. NIIP is based on a very "harmonized" view of the business world and it is too much focused on the US-based reality and interests. According to NIIP's concepts, all enterprises should work co-operatively by sharing all kinds of services, and resources, including humans. This approach is rather too generic and optimistic and probably, not in compliance with the current reality in most business sectors (NIIP, 1996). Therefore, although NIIP can be considered as a Reference Architecture to be considered before any new development in the VE area, it can not be easily adopted and deployed due to its generality and high level of abstraction. More specifically, the NIIP project developed concepts and prototypes for the static VEs. The selection of partners is performed manually, without using any type of matchmaking mechanisms, and as a consequence, the evolution of the VE could not be easily performed. The execution and management of shared business processes is done by shared, tightly coupled, business objects located in different physical and administrative locations. The interfaces among these business objects are static, pre-defined and well-agreed by the different partners. An alternative way for executing and managing business processes was the deployment of workflow management systems. In that case, the workflow management system is used for the management of internal business processes. The cross-organizational business process management is performed by the exchange of events generated and consumed by specialized gateways. In general, the NIIP project has not proposed so far a consistent approach for cross-organizational business process execution and management.

The X-CITTIC, Planning and Control Systems for Semiconductor Virtual Enterprises, is an Esprit funded project focused on VEs for the semiconductor industry (X-CITTIC, 1997). In this application domain, the manufacturing process is associated with sales order originated by a customer that can be located anywhere in the world. The management of sales orders can be accomplished through a globally distributed manufacturing network that can manufacture different pieces of the product on-demand. X-CITTIC expected to raise, to the virtual enterprise level, some of the techniques currently available in a modern shop floor (Veloso et al, 1998). Examples of such techniques are event-driven planning, scheduling, dispatching, and order release. The project also worked towards the direction of static VEs, where the establishment and configuration of the VE is performed manually and in a centralized manner. The execution of the shared business processes is done though special gateways that control the manufacturing control units (Adams & Dworkin, 2004). The management of shared process is performed in terms of events generated and consumed by the different partners. The semantic meaning of these events is tightly coupled with the business process that will handle the events. Events generated from one business process in one domain are forwarded in the domain's gateway (Debenham, 1998). The gateway locates the corresponding VE partner domain that will consume the event and forwards it to its gateway. The receiving gateway is responsible for the management of it by forwarding it to the internal process that will handle it. The relationships among the event consumers and providers are static and are not regulated or controlled by market driven approaches, like virtual marketplaces. The links between the different gateways are specified statically and could not be changed easily. Every domain pre-defines the events that can handle (Grefen et al, 1999).

The goals of MARVELOUS, an end user driven ESPRIT funded project, are the identification and harmonization of generic requirements for use of advanced IT in manufacturing and engineering across the maritime industry (MARVELOUS, 97). The project intended to guarantee consensus on requirements across the whole range of maritime users and to work closely with the technology providers in order to facilitate the formation of VEs. It also tried to ensure that the end-user requirements are feasible and can be translated into product developments. The project deployed open standards and distributed object-oriented technologies for the execution of business processes. The execution of shared processes is performed by specialized business objects, which are located in different partners (Cost, 1998). The relationships and integration among these entities is static and pre-defined leading directly to the concept of static VEs, while the business objects have tight coupled relationships among them. This means that no direct market oriented mechanisms are involved for the selection of partners. The integration of shared business processes is done in a manual and static way.

The VEGA project, Virtual Enterprise using Groupware tools and distributed Architecture, aims to establish an information infrastructure to support the technical and business requirements and operations of Virtual Enterprises (VEGA, 1998). Groupware tools and distributed architectures are being developed in compliance with product data standardization activities (STEP) and the current trends adopted by the forthcoming international industrial groupware specifications, for example the OMG (Zarli & Poyet, 1999). The approaches and developments resulting from a number of other ESPRIT projects were extended and the strategy for application integration by the distribution of a concurrent access to STEP databases were explored (Zarli & Poyet, 1999). A complementary route involves the design of a CORBA Access to STEP models (COAST) infrastructure to support the distribution of a product data by means of updated object broker technology. The VE partners are sharing production data stored in distributed federated databases managed by different domains. The main objective of VEGA was to provide a mechanism for sharing STEP oriented product designs across different domains for the manufacturing and production phase and thus, no major emphasis has been placed on the business process specification, execution, and management (Zarli & Poyet, 1999).

The PRODNET II project, Production Planning and Management in Virtual Enterprise, aimed at the design and development of an open platform to support industrial manufacturing VEs with special focus on the needs of Small and Medium

Enterprises (SMEs) (PRODNET II, 1998). The basic platform of PRODNET II includes, a Messaging System, for the exchanges of EDIFACT and STEP messages, a Co-ordination Module, for the execution of shared VE processes based on event management and CORBA remote requests, a Configurator, allowing the definition and parameterization of the VE and the behavior of each node, a distributed business process management system, that provides a proprietary first level coordination mechanism of business process execution at the VE level by supporting monitoring mechanisms, and finally a user driven partner search and selection mechanism without negotiation support based on public virtual marketplaces (Camarinha –Matos & Afsarmanesh, 1999a). The execution and management of shared business processes is based on message passing among distributed CORBA objects (Camarinha-Matos & Afsarmanesh, 1999b). The integration of shared business processes is pre-determined and based on user driven matchmaking services. The VE partners exchange EDIFACT messages for only electronic commerce purposes. The matchmaking service is a general-purpose directory service used to store company profiles related to certain processes and products. In general, the project does not address a generic mechanism for inter-domain business process execution and management. Additionally, the selection of partners in the VE is done in a manual and add-hoc way without negotiation process (Camarinha-Matos & Afsarmanesh, 1999c). Finally, the co-ordination of business process is done by the exchange of standard EDIFACT messages. The EDIFACT messages are only adequate for electronic commerce purposes and could not be applied for generic business processes.

The VENTO project, A Virtual Enterprise Organiser-Development of Advanced Groupware tools supporting synergy among enterprises in the emerging global market, aimed in the adaptation and integration of groupware tools to an integrated, inter-domain system that will operate in a distributed environment and will provide workgroups facilities and workflow management (VENTO, 1998). VENTO consists of a Workgroup Engine, which offers document management, history facilities and email functionality, Workflow Management System, providing with functions for workflow administration, process definition and process tracking, and Integration

Engine, establishing an object-oriented communication between workgroups and workflow and offering multilingual facilities. The VENTO platform is based on conventional, client-server, communication interactions. The business processes are specified using a business process definition language related to the workflow management system, while the execution of them is performed internally to each business domain and in a centralized way (Grefen et al, 1999). The workflow management system actually supports not the execution and management of shared business processes, but actually co-ordinates the execution of groupware services and the sharing of documents (Georgakopoulos, 1998). The coordination and management of shared groupware processes is performed by the exchange of proprietary messages based on TCI/IP protocol. The project deals directly with closed and well-integrated group of companies, i.e. static VEs that have static business relationships and tight coupling business processes. Additionally, the project does not specify any generic mechanism for inter-domain business process execution (Grefen et al, 1999). Finally, the project does not put emphasis on the dynamic selection of partners. On the contrary, the partners participated in the VE constitute a closed group of co-operating partners (Wognum et al, 1999a).

The GENIAL project, Global Engineering Network (GEN) Intelligent Access Libraries, aims in the establishment of a Common Semantic Infrastructure (CSI) (GENIAL, 1998). The CSI infrastructure enables enterprises from different business sectors to combine internal knowledge with engineering knowledge accessed on-line and world-wide via GEN services. The GENIAL platform consists of a framework for the systematization of engineering knowledge, i.e. generic software for the access, insertion, and administration of distributed engineering information and knowledge, and an electronic marketplace, which enables different companies to locate partners and establish co-operation with them. The project addresses only the establishment phase of VEs, i.e. the selection of partners and thus, no business process specification or execution mechanisms are provided (Fielding, 1998). The approach of the project is related to the sharing of information, e.g. industrial designs and modules, among different business domains. In general, it can be considered as a virtual marketplace or industry specific portal system for industrial modules and designs accessed by different industrial companies (Hunt et al, 1999). However, these domains neither share processes nor co-operate among each other, i.e. they do not explicitly constitute a virtual enterprise (Fielding, 1998). The execution and management of business processes is considered out of the scope of the project.

The VIVE project, Virtual Vertical Enterprises, aims at developing a general methodology that enables SMEs to exploit the opportunities of higher competitiveness offered by co-operative technologies (VIVE, 1999). The VIVE concept and its implementation is based on the development of robust methods for selecting and adapting information and communication technology solutions to enable the operation of such distributed business ventures and on the creation of a new entity, the “Business Integrator”. This new entity is capable of identifying market opportunities, specifying the required business process, and integrating the enterprise integration infrastructure in terms of communication and information. The VIVE concept leads to the static VE case where the integration of shared business processes is pre-determined and fixed, while the execution and management of shared business processes is achieved in a centralized way, i.e. the Business Integrator (Georgakopoulos, 1998; Zarli & Poyet, 1999). The VIVE concept neither provides any means of dynamic partner selection and negotiation nor loosely coupled business process execution (Fielding, 1998). The Business Integrator is actually a centralized node that undertakes the responsibility to co-ordinate and manages the relationships, i.e. the shared business processes among the partners. The co-ordination mechanism is based on integration of distributed objects, i.e. the Business Integrator plays the role of the information broker (Grefen et al, 1999). Virtual Enterprises is a rather new technology research area where a rapidly increasing number of projects and R&D activities are starting to consider it (Zarli & Poyet, 1999; Ouzounis, 1998c; Malone & Crowston, 1991; Georgakopoulos, 1998).

GLOBEMEN (Global Manufacturing and Engineering in Enterprise Networks) was an international research project under the IMS programme (IMS 99004). The IMS GLOBEMEN project started January 2000 and ran until the end of March 2003. The project consisted initially of 22 partners from 5 IMS regions but ended up with 19 partners from 4 IMS regions (Australia, EU, Japan and Switzerland). 11 of the partners were industrial partners of which the main part is within the one-of-a-kind production (OKP) representing industrial domains such as construction, process plant, shipbuilding, food processing, aerospace. The remaining industrial partners were ICT vendors representing the ERP and STEP domains. The additional 8 partners were research organizations or universities. The overall objectives of GLOBEMEN were: *"to define reference architecture for virtual manufacturing enterprises, to implement proof of concept industrial prototypes, to demonstrate core features of the architecture, and to promote deployment by IT vendors, manufacturing industry, academia and standardization"* (GLOBEMEN, 2002). In pursuing these objectives the GLOBEMEN project addressed three main business processes (GLOBEMEN, 2002): "Sales and Service", "Inter-enterprise Management", and "Distributed Engineering". Within each of these focus areas several subprojects were defined. Each of the industrial partners participated in a subproject within their own enterprise (one of the industrial partners was engaged in two projects). The aim of these sub-projects was to create prototypes for applications that could support identified issues related to the partner's involvement in doing business in an inter-enterprise environment. In total 11 prototype projects were developed each addressing different aspects related to preparation, setup and operation of virtual enterprises.

From the above description and analysis of the most influential projects in the area of Virtual Enterprises, certain conclusions can be drawn:

- Most of the emerging R&D projects and scientific activities have different conception, definition and interpretation of the term virtual enterprise. A clear definition and distinction of the VE model in comparison with supply chain management, virtual organization and extended enterprise concept is still missing.
- Most of the projects did not even consider the major distinction among static and dynamic VEs,

- Most of the projects are analyzing, designing, and developing solutions for static VEs, i.e for pre-defined number of partners with fixed business process interfaces among them and static proprietary co-ordination mechanisms,
- Most of the projects have as a selected business sector the manufacturing area where the coordination and management of processes is tight coupling, while the customization and integration of shared business processes is static, manual and pre-defined. Therefore, the business relationships among the partners are rather medium to long term and consequently, the static VE model is more suitable.

3.2.8 Breeding environments

One of the challenges related with the dynamic nature of virtual enterprises where companies get together as an answer to a specific customer need is that the participating companies would not necessarily be very familiar with each other. There can be a lot of challenges related with new partners: including trust (Handy, 1995; Sabherwal, 2003; McGrath & More, 2003; Berg & Lieshout, 2001) as well as consideration regarding uncertainty of their performance and commitment.

Another related challenge is the time related to the setup of the virtual enterprise:

> *"The success of the virtual organization model is tied to the ability of 'real' companies to form virtual organizations rapidly to meet an emerging time-based opportunity."* (Goldman et al, 1995)

The temporary nature sets up requirements to make the creation and operation of virtual enterprises efficient and sufficiently reliable (Ollus et al, 2003). A way to come around this is through establishing more long term relationships such as enterprise networks and through establishing a preparedness to set up and operate virtual enterprises. Different terms with the same basic idea are used for this type of organizational form including: Network / Virtual Web / Source Network / Support Network / Breeding Environment. As a means to facilitate a rapid forming of virtual organizations Goldman et al introduces what they call a Virtual web:

"The web is an open-ended collection of prequalified partners that agree to form a pool of potential members of virtual organizations." (Goldman et al, 1995)

The Virtual Organization Cluster consortium (VOSTER), funded by the EU's 5th framework program, with the aim *"to collect, analyze and synthesize the results from a number of leading European research projects on Virtual Organization (VO)"* VOSTER defines Network / Source network / Support network / Breeding environment as one of the core organization concepts accordingly:

"A more stationary, though not static, group of organizational entities which has developed a preparedness to cooperate in case of a specific task / customer demand." (VOSTER, 2003)

The dynamic network of Miles and Snow contains similar characteristics:

"Its major components can be assembled and reassembled in order to meet complex and changing competitive conditions" (Miles et al, 2000)

The basic idea behind the network in comparison to the virtual enterprise is based upon more stable and long-term relationships among the partners. This does not mean that the network keeps status quo. On the contrary the network should continue to evolve including new partners possessing new attractive competencies. This should be done both to complement and extend the network's competencies but also in some situations to replace or supplement the existing pool of competencies.

As mentioned trust related issues can be challenges in relation to inclusion of new and unknown partners in a network. A way to reduce the trust related barriers is through an assessment of the partner. The assessment can cover both a technical conformity as well as a functional conformity of the potential partner (Berg & Tolle, 2001). *Technical conformity* addresses issues such as:

> *"Do the companies' respective systems comply with the same ICT-standards, so they can exchange data?"* (Berg & Tolle, 2001)

Assessing for technical conformity will not be sufficient in order to include the partner in a virtual enterprise. In many cases a more elaborate assessment is needed. That is, in order to really exploit the opportunities of enterprise integration the *functional conformity* should also be assessed addressing issues such as:

> *"Does a common view exist on the inter-organizational activities and do business processes match (inputs/outputs required, processing sequence, lead times)?"* (Berg & Tolle, 2001)

The assessment can be performed by different entities either by a partner participating in the network or by an outsider such as e.g. a 3rd party assessor. The idea behind 3rd party assessment is that an independent and trusted 3rd party, such as a certified public accountant (CPA), assesses if a potential partner complies with a set of general best practices or a more specific set of requirements as e.g. financial requirements. Thus, if a trusted assessor approves a potential partner the potential partner obtains a reference based trust (Berg & Lieshout, 2001). This is similar to the e.g. ISO9000 certifications that indicates that a company is operating in accordance with the quality standard.

So far 3rd party assessment is most widely applied in relation to the business-to-consumer ecommerce on the Internet (Berg & Tolle, 2001). This includes companies such as CPA WebTrust that issues web-seals which:

> *"illustrates that a Web site has been examined by a qualified CPA who has independently verified that the site complies with a series of WebTrust e-business best practices"*

3rd party assessment in relation to business-to-business such as within a network or virtual enterprise is still to be seen. Compared to business-to-consumer, assessment of partners for virtual enterprise and networks is more comprehensive. In business-to-consumer the consumers' primary concerns are related to if the goods are what it seems to be and not least concerned if the receiver of the credit card information is trustworthy.

Assessment of potential partners in a business-to-business environment (e.g. enterprise network or virtual enterprise) includes both the technical as well as the functional conformity described above. That is, one aspect is to assess if the partners technical systems can be integrated, another element is to assess if the partners share a common view of the inter-enterprise activities and if their business processes match or can be aligned. The need for a more comprehensive assessment of potential virtual enterprise partners means that on one hand some companies can assess it to be important to leave for outsiders. On the other side however:

> *"The ideal of VEs is driven by the strategic desire to concentrate on one's core business. From this perspective it is not sensible to have a large in-house assessment function."* (Berg & Tolle, 2001)

Thus, in the long run one could foresee that 3rd party assessment will evolve as a respond to the need related to the new business requirements previously described. Some elements of the assessment could become a sort of prerequisite for being considered as a potential partner. That is, similar to e.g. ISO9000 certification potential partners themselves have to document that they comply with a given standard or set of best practices. This would put the task and the cost related to the assessment on the potential partner. Another reason that speaks for 3rd party assessor is that a certain level of expertise is required to carry out an assessment. Different types of assessors could be foreseen:

> *"Assessment should be left to specialized third parties. At this moment certified public accountants (CPAs) are most advanced in this area. Other*

professionals, e.g. EDP-auditors [Electronic Data Processing, ed.] and even public notaries or banks could play a role in certifying enterprises which meet certain thresholds to participate in VEs." (Berg & Tolle, 2001)

Although part of the assessment could be outsourced to 3rd parties some part of the assessment will still be left to the partners in the network.

3.2.8.1 Real examples of breeding environments

The three examples of breeding environments below represent regional cluster networks where the companies have joined together to support customer acquisition and delivery.

- **HarmaGlobal:** A study called HarmaGlobal aimed to clarify the e-business and networking readiness of SME manufacturing oriented metal industry companies in Finland. The main development areas that came up in these terms are summarized in Table 3.8.
- **Net A:** "Net A" consists of 17 high tech enterprises in the Varkaus region in the eastern part of Finland (Ollus et al, 1999). The main focus areas are: automation and instrumentation technology, electronics, precision mechanics and software. In addition they offer design, training and consultancy in the field. Together they form a network or breeding environment enabling turn-key deliveries as a large company, but benefiting from the flexibility efficiency of the small partner companies. The "Net A" was established in 1992 after a very turbulent period in the region. The region has a strong background in the pulp and paper manufacturing and manufacturing of energy production machinery. A considerable amount of SMEs supported the activities. During the recession in the late 80's and early 90's the large industries in the area went through a major restructuring process.

  Three large American companies (Honeywell, Foster & Wheeler and CEC) became owners of most of the manufacturing capacity in the area. Only the paper production remained as Finnish owned. To answer the threats the SMEs in the area, which earlier had supported the manufacturing industry decided to collaborate and join their efforts to survive during the turbulent era. In ten years the number of employees in the member companies has grown from 100 to about 300. From local service providers the companies have also become international and can offer services world-wide. The collaboration is based on common agreements for collaboration. At the same time they have maintained flexibility. They have learned to collaborate in the network and market each others' capacity. The member companies can also benefit from acquiring common services as a "large" company.

Table 3.8 Main development areas of Harmaglobal (Salkari et al, 2002)

Processes	
Development area	**Development challenges and Lessons Learnt**
- Communication of forecast data to suppliers	- Focusing the data, selecting the right data - Communication format - N-tier suppliers: it is not enough if forecasting data reaches only the 1st tier - The business culture is not open for sharing forecast information
- Communication of capacity information	- Business environment not open for sharing capacity information - Capacity is monitored by rules of thumb – even internally it is not managed with IT systems
- Cost and profit-sharing in processes that cross company boarders	- Fair play promotes eBusiness ("win-win") - ICT or process development must not be used to outsource costs without paying for it
- Common R&D information database / product information database	- Unwillingness to share R&D information. Good relations and trust between companies needed. Requires long term companionship
ICT	
Development area	**Development challenges and Lessons Learnt**
- Integration of ERP systems	- In many cases requires renewing or updating the existing ERPs - ERP providers are not situated close by
- Increasing use and usability of existing tools in managing network operations (For example forecast data can be sent by e-mail)	- The communicated data must be well focused and in the right format - Frequency of data communication is limited - Processes must be simple
- Capacity and warehouse management systems in network (it can be studied who has the shortest delivery time etc.)	- Systems too inaccurate for SMEs. Warehouse and workload can be managed more accurately by rules of thumb since the business is so small
- "Lean" Internet –based ERP-systems between principals and suppliers - Means Extranet solutions and where appropriate, use of homepages and e-mail for information exchange and management	- Related development of business processes and practices (usually this plays an important role because a lean system can be used most effectively when some good practices form a part of the system. This also increases the flexibility of the system)
- Common portal for suppliers	- Different processes and products. Portals need tailoring - Portals do not integrate information, people still have to copy information from one system to another - Updating the portal? Whose responsibility, who is ready to bear the costs?
- Communication of forecast and order data (with existing systems and standards)	- Many different systems and standards – interoperability of these systems - Many parallel systems → expensive for SMEs - EDI connections are seen as a possibility but also quite expensive and inflexible – especially from the suppliers' point of view.

- **Virtuelle Fabrik:** Virtuelle Fabrik is another example of a network of SMEs that acts as a VE breeding environment, within which dynamic VEs are created whenever a business opportunity is found. From the experience of operation of this network a number of problems requiring advanced research and or new developments have been identified, as summarized in the Table 3.9.

Table 3.9 Problems requiring new developments

VO Marketing: Marketing organisation (sales potential and sales territory) Advertising, Public Relation Appearance as a community -To offer complete / total products and equipments (not one or more companies together) Installation and removal of a stand at fairs / exhibitions; stand-costs, personnel costs… VO Customer: **Customer benefits – a customer of the VO, not a customer of a VO partner** Customer confidence -Who is the person that is responsible /that the customer can trust Competence and customer-satisfaction -Comparable with our competitors Responsibility before the customer – Costs, Appointed time, Quality, Guaranty, warranty, product-liability, after-sales- service Partner of sales-contract – Who is the legitimate contract-partner. Internal Operation -competence & cooperation = "cooperation" Confidence between Partners (TRUST) •New ideas, passing information and knowledge •Quality of performed jobs •Appointed time •Terms of contract (VO-customer; you can't ask each partner) **Responsibility** •Each partner feels himself responsible? (each partner has one person who is responsible for topics of VO) •Position, status and influence in his company and VO •Qualification, professional experience from the staff in each company **Aims/ Goals** •What is the goal of each company to be member in a VO •Motivation to do two jobs (as an individual company and as member of a network) **Rules** •Resources, commitment, intercession of possible customers •Responsible person for VO-topics in each company •All rules must be fixed; confidence that the rules will observed and valid •Build of VO company-culture

3.2.9 Summary of new enterprise forms

As a respond to the new business challenges described in the previous section 3.2 the virtual enterprise concept, the related concept of extended enterprises and breeding environments has been described in this section.

The virtual enterprise concept that will be applied in this research is aligned with the general understanding of the virtual enterprise in the literature. The two core elements in this understanding are the agile and usually short-term virtual enterprise that is set up and prepared by a more long-term enterprise network.

A short definition of a virtual enterprise will be applied in this dissertation:

> *"A Virtual Enterprise is a customer solution delivery system created by a temporary and re-configurable ICT enabled aggregation of core competencies."* (Sari et al, 2004, 2005a & 2005b)

Compared to the virtual enterprise the extended enterprise is of a more stable nature. The notation of extended enterprises is usually used to describe more long-term relationships while virtual enterprises are short-term business structures with a high degree of flexibility. The breeding environments, is a long-term network of partners from which temporary virtual enterprise can be configured. Thus, the enterprise network possesses characteristics similar to the extended enterprise. The enterprise network, is seen as a means to bridge the gap between *the theoretical* but rarely seen in its purest form virtual enterprise configured on the spot with previously unknown partners as an answer to a specific customer need, and then *the present situation* of many companies which are engaged in long-term relationships with trusted partners from which they can deliver customer solutions.

3.3 Artificial Neural Networks (ANN)

3.3.1 Why ANN?

Neural network can be useful when rules are not known, either because the topic is complex or because no human expert is available. If training data can be generated, the system may be able to learn enough information to function as well as or better than an expert system. This approach also has the benefit of ease of modification by retraining with an updated data set, thus eliminating programming changes and rule reconstruction. The data-driven aspect of neural networks allows a system adjustment as a result of changing environments and events. Another advantage of neural networks is the speed of operation after the network is trained. Neural networks can be one of the best solutions for some of the problems that have proven difficult for expert system developers, and can allow them to address problems not amenable to either approach alone. Neural networks have the potential to provide some of the human characteristics of problem solving that are difficult to simulate using the logical, analytical techniques of expert system and standard software technologies. For example, neural networks can analyze large quantities of data to establish patterns and characteristics in situations where rules are not known and can, in some cases make sense of incomplete data.

3.3.2 Introduction to ANN?

Artificial Neural Networks (ANNs) are biologically inspired models analogue to the basic functions of biological neurons. They have a natural propensity for storing experiential knowledge, and resemble the human brain in the sense that training rather than programming is used to acquire knowledge.

A neural network consists of a number of nodes massively interconnected through connections. The nodes are arranged in layers: an input layer, an output layer, and several hidden layers. The number of hidden layers depends on the type of problem. The nodes of the input layer receive information as input patterns, and then transform the information through the connections to the other connected nodes layer by layer to the output layer nodes. The transformation behavior of the network depends on the structure of the network and the weights of the connections. Figure 3.3 shows a multi-layered neural network.

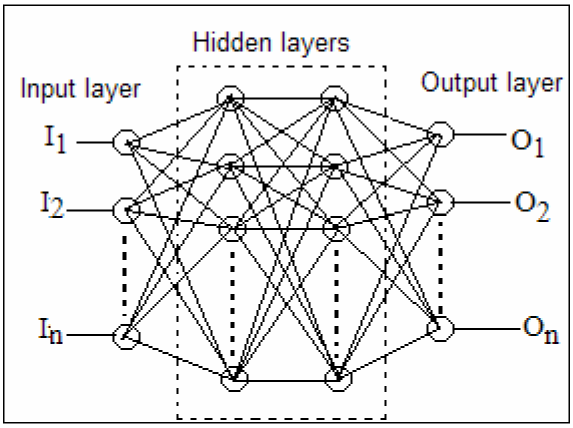


Figure 3.3 Multi-layered neural network

3.3.3 Application of ANN in VE

In the past, the focus of manufacturing sector was on supplier partnerships to improve cost and quality (Ellram & Cooper, 1990). In today's ever-changed markets, the focus has shifted to speed, flexibility and innovation, putting more emphasis on shortened product development cycle, integrated enterprise modeling, improved outbound logistics and fast response to market needs (Desbarats, 1999). Recent years have seen significant changes made in terms of the manufacturing paradigm shift from traditional manufacturing to a world of agile and virtual manufacturing, capitalizing on the advances in information and Internet technologies. The benefits of adopting an Internet approach can be demonstrated through the improvement of manufacturing activities in various aspects of the value chain activities from product and process design to marketing (Dell, 1999).

To many companies, the Internet provides a global link to the company's customers and suppliers. With the rapid advances of electronic and communication technologies, a VE that shares the information and resources among loosely connected firms can be realized. The idea of forming a VE is meant to establish a dynamic organization by the synergetic combination of dissimilar companies with different core competencies, thereby forming an organization to perform a given business project to achieve maximum degree of customer satisfaction (Goranson, 1999). Customers and suppliers can directly access the company almost at any place and at any time in the world for acquiring useful information (Lee & Lau, 1999). From the commercial point of view, doing business via the Internet can drastically reduce transaction and information cost of a business and provide companies with an opportunity to increase productivity through the smart use of information. The exchange of information in business to business commerce requires the alignment of different data platforms to obtain meaningful information (Pawar & Driva, 2000). However, the emergence of the need to use available information smartly requires a redesign of operational infrastructures to change the requirements of the users in using information to do their jobs. When a VE becomes even bigger and more complex, critical and hard decisions have to be timely and made accurately. As a result, the role of decision-support functionality becomes increasingly essential, and it is very likely that it is going to become a key part of the infrastructure of an enterprise, which aims to enable the decision-makers to cope with a continuously changing and unpredictable environment.

The empirical research in the area of supplier selection mainly focused on studying the relative importance of various partner attributes such as price, quality, and delivery performance (Dickson, 1966). There are some well-known studies in this area. For example, Petroni & Braglia (2000) concluded that the relative importance assigned to a partner attribute was primarily based on the type of risk involved in a specific purchasing situation. Purdy & Safayeni (2000) suggested that management would generally be willing to pay 4–6% higher than the lowest acceptable bid if the product performance is superior. Based on a comprehensive review of partner evaluation methods, Sarkis & Talluri (2002) concluded that quality was the highest ranked factor followed by delivery performance and cost. It can be concluded from these studies that partner selection decisions must not be exclusively based on least cost, delivery, quality criteria and that also a critical factor such as partner past performance need to be incorporated into the evaluation and selection process.

Decision problem of assessing the partner companies' performance often depends on experience of managers. It is necessary to develop a computer-aided decision system in performance evaluation to reduce dependence on personal experience, improve decision quality and shorten response time. Here, convention computer techniques are somehow inadequate. Artificial neural networks (ANN) have advantages over traditional classification methods such as discriminate analysis. Neural networks however are being utilized for those processes that often lack structure and organization and therefore the right vehicle for such evaluation analysis.

3.4 Analytic Hierarchy Process (AHP)

3.4.1 Why AHP?

The Analytic Hierarchy Process was developed by Thomas Saaty [17] in early 1970s. The strength of the AHP approach lies in its ability to structure a complex, multiattribute, multiperson, and multiperiod problem hierarchically. In addition, it can also handle both qualitative and quantitative attributes. Pairwise comparisons of the elements (usually, alternatives and attributes) can be established using a scale indicating the strength with which one element dominates another with respect to a higher level element. This scaling process can be translated into priority weights (scores) for comparison of alternatives.

The advantages of using AHP, in the partner selection are listed below:

(i) The partner selection by AHP takes into consideration both the quantitative and qualitative factors.
(ii) AHP can display complicated selection factor in simple concepts of hierarchy, which can be accepted easily by a decision maker.
(iii) AHP goes through a dynamic group discussion and denotes the priority of a decision with certain numerical values. It does not involve statistics or probability theory, thus giving the user a better sense of reality.
(iv) AHP involves group discussion and dynamic adjustment to finally achieve the consensus. The evaluation is conducted by the participating experts who decide jointly on the parameters for pairwise comparison. It is thus more of a qualitative analysis.
(v) Non-quantified elements, after group evaluation and a mathematical process can be quantified by numerical values to indicate a decision's priority. A decision maker can reach the choice of partner in a very short time without resorting to precise data.

3.4.2 Introduction to AHP

The general approach followed in AHP is to decompose the problem and to make pairwise comparison of all the elements (attributes, alternatives) at a given level with respect to the related elements in the level just above. AHP consists of three stages of problem solving decompositions, comparative judgements and synthesis of priorities.

The degree of preference of the decision maker in the choice for each pairwise comparison is quantified on a scale of 1 to 9, and these quantities are placed in a matrix. A preference of 1 indicates equality between two items while a preference of 9 (absolute importance) indicates that one item is 9 times larger or more important than the one to which it is being compared. This scale was originally chosen, because in this way comparisons are being made within a limited range where perception is sensitive enough to make a distinction.

Table 3.10 Interpretation of Entries in a Pairwise Comparison Matrix

Value of aij	**Interpretation**
1	Objectives i and j are equal of importance
3	Objective i is weakly more important than objective j
5	Experience and judgement indicate that objective i is strongly more important than objective j
7	Objective is very strongly or demonstrably more important objective j
9	Objective i is absolutely more important than objective j
2, 4, 6, 8	Intermediate values-for example, a value of 8 means that objective i is midway between strongly and absolutely more important than objective j

AHP usually involves three stages of problem solving. These are the principles of decomposition, comparative judgments, and synthesis of priority. The decomposition principle calls for constructing a hierarchy or network to represent a decision problem. The overall objective is located at the top of the hierarchy, and the criteria, sub criteria, and alternatives are placed at each descending level of the hierarchy. To apply the principle of comparative judgment, the users set up a comparison matrix at each level by comparing pairs of criteria, or pairs of alternatives at the lowest level according to Table 3.10. Once the matrix of pairwise comparisons has been developed, one can estimate the relative priority for each of the alternatives in terms of the specific criteria. Preferences derived from a criteria or sub criteria matrix are used to calculate a composite weight for each alternative. This part of AHP is referred to as synthesis. This enables AHP to obtain not only the rank order of the alternatives, but also their relative standings measured on a ratio scale. The alternative with the highest overall rating is usually chosen as a final solution.

3.4.3 Application of AHP in VE

Some VEs may be long term alliances that last for an indefinite number of business processes. Other VEs may be established for a single business opportunity, as pointed out by Camarinha-Matos and Afsarmanesh (1999), this is perhaps the most typical kind of VE. The goal of VE is to meet the challenge of customized manufacturing or customer driven manufacturing. Customer driven manufacturing will ultimately lead to one of a kind production (Jagdev & Browne, 1998; Wortmann, 1992). Thus, even for a long term alliance VE, the products manufactured are changed frequently and some partners are changed too. In this way, reconfiguration is done frequently, and this is a great challenge to the VE practice. Therefore, effective techniques for selecting partners in VE practice are essential.

In selecting the partners for a business opportunity in a VE, there are many factors to be taken into consideration. These factors include cost, quality, trust, credit, delivery time and reliability. However, the key factors to be addressed are cost and time. As pointed out by Jagdev and Browne (1998), high quality products are a necessary precondition for entry into the market and for many manufacturers high quality are no longer the basis of competitive advantage, so cost and time to market are the basis of competitive advantage. In doing research for the PRODNET project, Camarinha-Matos & Cardoso (1999) present a framework for partner selection and describe the functionalities in detail, but no techniques to make the tradeoff based on the cost and time are proposed. Under VE and supply chain management, the partner selection problem is studied by Gunasekaran (1998) and Maloni & Benton (1997), and they point out that the mathematical models and optimization methods are still a challenge. The partner selection problem is also studied under project management in the cooperation relationship of subprojects contracted by partners (Brucker et al, 1999; Elmaghraby, 2000; Wang et al, 2001). In the study of Brucker et al. (1999), the partner selection is embedded in the project scheduling problem.

In Wang et al.'s (2001) study, the costs and completion time of the subprojects bidden by the candidates are taken into consideration and a mathematical programming model is presented to model the problem, and the problem is solved by a genetic algorithm. In a VE environment for manufacturing, the partners are geographically distributed, so besides the cost and time required for performing the tasks by the partners, material transportation between partners consume money and time too. Such cost and time are significant enough not to be ignored. With the transportation cost and time considered, the partner selection problem is much more complicated. Taking the processing cost and the transportation cost into account, Wu et al. (2003) modeled the partner selection problem by a network model and an efficient algorithm was presented to solve it. However, in that model, the time factor is neglected.

Several multi-criteria, mathematical programming, and other advanced methodologies have also been proposed for partner selection. Some of these methods are weighted linear models, linear programming, analytic hierarchy process, data envelopment analysis, neural networks etc. Although several effective techniques and models have been utilized for evaluating partners, there is little work in incorporating variability measures into the evaluation process. Clearly, integration of performance variability into the decision making process, and the identification of effective alternative choices provides the buyer with flexibility in the final selection process. Different from other researches, we propose the task price, caution price, task completion probability and the partners' performance as the criteria to be considered in the partner selection process.

3.5 Program Evaluation Review Technique (PERT) Analysis

3.5.1 Why PERT?

PERT (Chen and Chang, 2001; Dubois et al, 2003; Fatemi Ghomi and Rabbani, 2003) is the most widely used management technique for planning and coordinating large scale projects. By using PERT, managers are able to obtain (Stevenson, 2002):

1. A graphical display of project activities,
2. An estimate of how long the project will take,
3. An indication of which activities are the most critical to timely project completion,
4. An indication of how long any activity can be delayed without delaying the project.

Therefore, PERT is suitable for computing the order-completion time and forecasting the order fulfillment ability in the supply-chain system.

3.5.2 Introduction to PERT

PERT, the Project Evaluation and Review Technique, is a network based aid for planning and scheduling the many interrelated tasks in a large and complex project. It was developed during the design and construction of the Polamaris submarine in the USA in the 1950s, which was one of the most complex tasks ever attempted at the time. Nowadays PERT techniques are routinely used in any large project such as software development, building construction, etc. Supporting software such as Microsoft Project, among others, is readily available. It may seem odd that PERT appears in a book on optimization, but it is frequently necessary to optimize time and resource constrained systems, and the basic ideas of PERT help to organize such an optimization.

PERT uses a network representation to capture the precedence or parallel relationships among the tasks in the project. As an example of a precedence relationship, the frame of a house must first be constructed before the roof can go on. On the other hand, some activities can happen in parallel: the electrical system can be installed by one crew at the same time as the plumbing system is installed by a second crew.

The PERT formalism has these elements and rules:

- Directed arcs represent *activities*, each of which has a specified *duration*. This is the “activity on arc” formalism; there is also less-common “activity on node” formalism. Note that activities are considered to be uninterruptible once started.
- Nodes are events or points in time.
- The activities (arcs) leaving a node cannot begin until all of the activities (arcs) entering a node are completed. This is how precedence is shown. You can also think of the node as enforcing a rendezvous: no one can leave until everyone has arrived.
- There is a single starting node which has only outflow arcs, and a single ending node that has only inflow arcs.
- There are no cycles in the network. You can see the difficulty here. If an outflow activity cannot begin until all of the inflow activities have been completed, a cycle means that the system can never get started.

Consider the example given in Figure 3.4. Perhaps the pouring of the concrete foundation (activity A-B), happens at the same time as the pre-assembly of the roof trusses (activity A-D). However, the finalization of the roof (activity D-E), cannot begin until both A-D and B-D (assembly of the house frame), are done. Of course B-D cannot start until the concrete foundation has been poured (A-B). All of this precedence and parallelism information is neatly captured in the PERT diagram.

There are two major questions about any project:

- What is the shortest time for completion of the project?
- Which activities must be completed on time in order for the project to finish in the shortest possible time? These activities constitute the critical path through the PERT diagram.

The process of finding the critical path answers the first question as well as the second. Of course we need to know how long each individual activity will take in order to answer these questions. This is why the arcs in Figure 3.4 are labeled with numbers; the numbers show the amount of time that each activity is expected to take (in days).

The critical path is of great interest to project managers. The activities on the critical path are the ones which absolutely must be done on time in order for the whole project to be completed on time. If any of the activities on the critical path are late, then the entire project will finish late! For this reason, the critical path activities receive the greatest attention from management. The non-critical activities have some leeway to be late without affecting the overall project completion time.

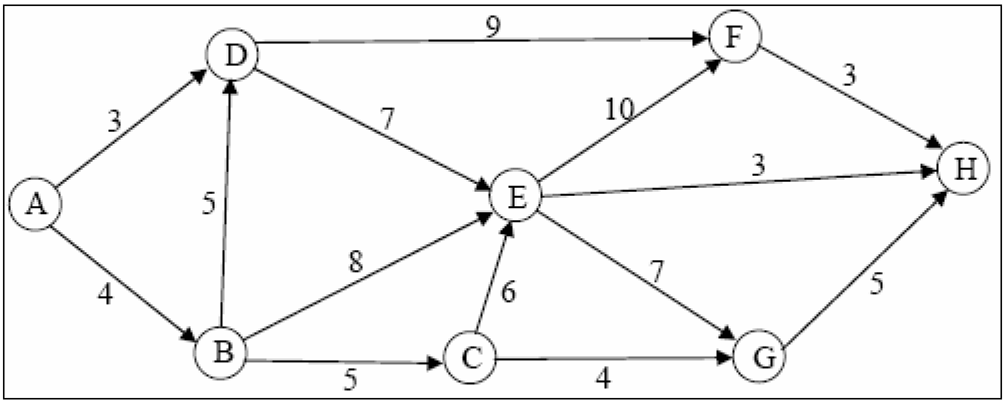


Figure 3.4 An example of a PERT diagram

CHAPTER 4

VE BUSINESS MODEL

4.1 Introduction

The purpose of this chapter is to address the first research question: *RQ1: What is a VE?* Being the pivotal point of this research virtual enterprises have been addressed throughout the complete duration of the research. That is, every time a presentation has been given or that a publication has been written the basis virtual enterprise concept have been presented.

In the following section 4.2, the virtual enterprise concept will be summarized based upon the appended publications. In the subsequent section 4.3 life cycle model of the VE will be described

4.2 Concept of the Virtual Enterprise

4.2.1 Summary of result

The literature review in chapter 3 revealed that the concept of a virtual enterprise has been devoted attention from numerous authors, focusing on various aspects of a virtual enterprise. This research builds upon the virtual enterprise concept as defined in the following way:

> *"A Virtual Enterprise is a customer solution delivery system created by a temporary and re-configurable ICT enabled aggregation of core competencies."* (Sari et al, 2004)

VEs can be formed to perform a one-of-a-kind production or service task, such as building a plant or building a ship. Alternatively, VEs can also be formed to perform the manufacturing of a product line or the delivery of after sales services for a product line (Hartel & Burger, 2001). In this latter case, several VEs can be set up depending on, e.g., the service needs.

Our view on virtual enterprises is illustrated in Figure 4-1. The figure shows that a group of enterprises comes together to create a more or less formalized and prepared network. This network then serves as the breeding ground for setting up specific VEs based upon identified customer needs and requests.

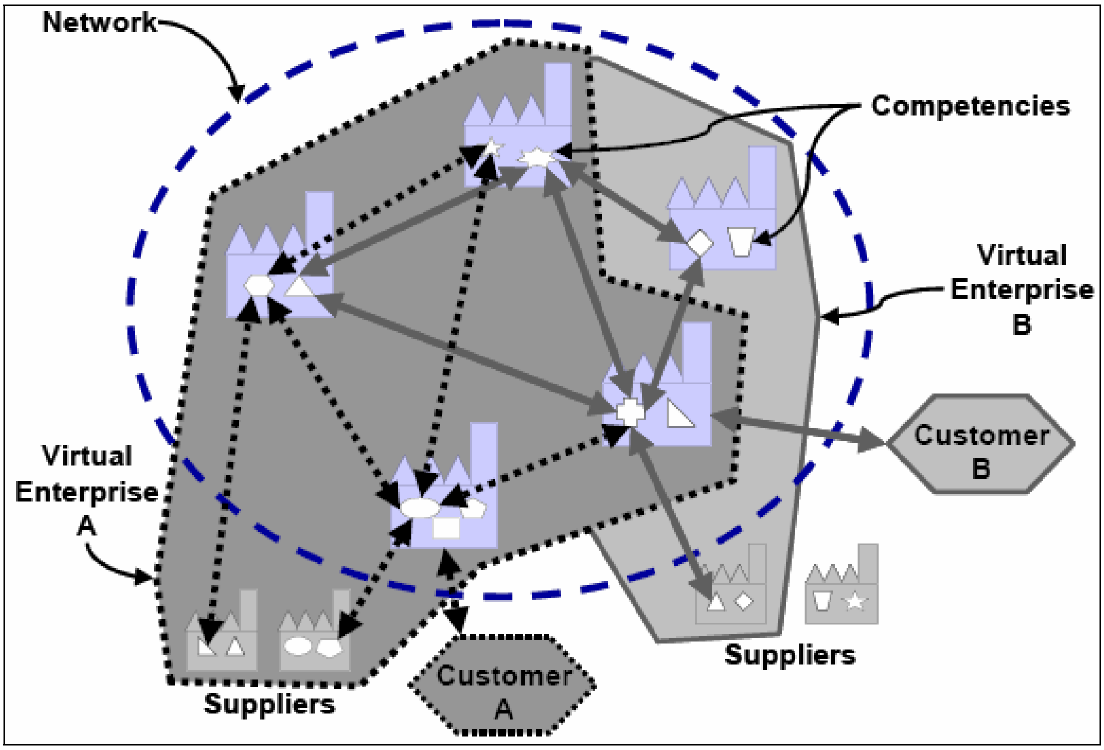


Figure 4.1 Enterprise network and virtual enterprise

Thus, a Virtual Enterprise is a temporary, co-operative association of enterprises, which is established as a separate business entity to share capabilities, capacities, cost and risk in fulfilling specific customer demands. Formation of the VE materializes through a customer focused configuration of the core competencies available in the network and possibly through inclusion of additional required competencies provided by non-network member companies. Though being comprised by competencies from various partners, the VE appears to the customer as one, unified, and attuned enterprise; hence its virtual nature as an enterprise. Accordingly, the business processes are not carried out by a single enterprise; rather every enterprise is just a node in the VE that adds some value to the product.

The main purpose of a network is to prepare, set up and manage the life cycle of VEs and to prepare product life cycles. The network is thus to be seen as a potential from which different VEs can be established in order to satisfy diverse customer demands. The network will seek out and/or await customer demands, and when a specific customer demand is identified the business potential is realized by forming a VE. Compared to a virtual enterprise a network can accordingly be perceived as a relatively long-term co-operation since it typically sets up multiple VEs. Conversely, the VEs are of a temporary nature to be decommissioned when an OKP product has been delivered.

A network is created based on core competencies assigned from different partners of the network. The network can therefore be characterized as a portfolio of core competencies that are available to realize VEs and products. This competence portfolio is dynamic in the sense that competencies can leave and join the network. In addition, a network can be characterized as a product-oriented network focusing on the strategically important, value adding partner competencies in the potential supply chains, while typically excluding off-the-shelf suppliers. A network thus constitutes a crucial subset of a wider set of supply chain networks.

4.2.2 Challenges in enterprise networks and virtual enterprises

One of the big challenges in relation to enterprise networks and virtual enterprise is to be able to configure/set up the VEs within a competitive timeframe. To put it simply: one thing is to be able to compose a virtual enterprise based upon the best available competencies, but if the setup cannot be done within a competitive timeframe it is not relevant. Thus, there is a need for collecting and structuring the knowledge that can facilitate the setup and operation of enterprise networks and their accompanying virtual enterprises. System framework & structured methodology and the accompanying Virtual Enterprise Methodology (VEM) presented in the following chapters have been developed as an attempt to overcome this challenge. An important means for speeding up the setup and operation of networks and VEs as well as to get an efficient and reliable operation is by applying well tested and proven models and systems as well as preparing additional ones. Thus, one of the important decisions one has to take when considering setting up enterprise network(s) and VEs is to determine the type and degree of work preparation.

4.3 Life Cycle Model of the Dynamic Virtual Enterprise

A life-cycle model usually describes the key phases and activities required during the existence of an entity. According to ISO (ISO 91 and 94), a life cycle can be defined as “the finite steps a system may go through over its entire life history. The different life cycle phases define types of activities which are pertinent during the life cycle of the entity”. In our case, the VE life-cycle model consists of three key phases that should be followed for the establishment and management of a VE. Every phase consists of more specific steps that describe the main operations that should be done by different human roles from technical point of view. It should be noted that the following life-cycle model is best applied in the DVE model that is the core work of this dissertation.

In order to better understand the life-cycle model, the following scenario is provided. A Company called *Online-Book* sells books to customers online. Part of the book-selling business process is the distribution process, i.e. the delivery of the book to the customer. Online-Book has not a particular way to distribute the books and looks for a logistic partner to outsource this process. The Online Book company knows exactly the properties and attributes of the distribution process. In order to find potential partners with logistic capabilities, On-Line Books deploys a third party virtual marketplace or enterprise network that provides matchmaking services for logistic companies.

The network specified several logistic process templates for different logistic services. One of the logistic process templates is the book delivery process. This process template has, for example, as properties the destination, the price for the delivery, the payment method, when the payment should be done, delivery day, the guaranty in case of problems, etc. Logistic companies, that can deliver books, use the standard book delivery process template and register their process offerings into a particular network by specifying their terms and conditions.

When a customer orders a book from the Online-Book, the company searches the company network, locates the potential logistic partners, negotiates with them about the price, location, delivery day, time, quantity, etc. and selects the most suitable one. The selection of the partner is highly associated with the characteristics of the customer, i.e. his location and preferences. Then, the Online-Book uses the logistic process provided by the selected partner and serves the customer. When a new customer comes and places a book order into the shopping system of the company, the company uses again the network and locates, negotiates, and selects probably another logistic company that can better satisfy the requirements of the new customer.

The above scenario is a very typical, though simplistic, one that reveals most of the characteristics of the DVEs concept. In order to support a scenario like this, a management platform for the management of DVEs is required (Ouzounis & Tschammer, 2000; Stricker et al, 2000; Camarinha-Matos & Afsarmanesh, 2003). In more general terms, the following definitions and concepts are provided in relation to the life cycle model and this thesis specifically.

In general, a VE is a set of business domains that jointly and dynamically co-operate to provide value-added services to a customer in a transparent way, i.e. the customer does not know about the existence of the different business domains involved in the service provision. A business domain is an administrative domain that poses its own resources, infrastructure, and services and imposes its own restrictions and regulations on them in terms of access control and authentication. Business domains jointly co-operate by sharing services, i.e. one business domain deploys services provided by one or more other business domains in a consistent and well-regulated way. Every request for a service deployment from one domain is checked for permission by the requested domain. The access control and authorization of requests before the service provision is based on a temporary contract that has been agreed by both domains, i.e. by both the requestor and supplier. The business domain that requests a service by one domain is called the requestor while the domain that provides the service is called the supplier.

The technical representation of a service is a business process. The business process can be either, local or remote. All the processes provided in self-contained manner by this domain are called local processes. Therefore, a domain has full control of its own local processes and can impose any type of access control constraints. On the contrary, when a process could not be provided by one particular domain, but should be deployed by a remote one, is called remote.

The domain that provides a service, i.e. a business process, directly to a customer is called the VE representative/coordinator/administrator. The process that is provided by the VE representative to the customer is called VE process. The VE representative domain represents the VE in the outside world in a similar way like a normal company. The domains that participate in the provision of VE processes in the context of the VE are called VE partners. Initially, when the different business domains have no relationships among them, i.e. they do not share any processes, are called VE candidate partners or network members. A VE candidate partner is becoming VE partner after a negotiation process that involves the potential requestor of the process and the potential supplier of the process. When an agreement is reached then the potential supplier becomes a VE partner. This negotiation process is done dynamically and during the provision of VE process to the customer. The agreement might last for only one business process deployment or for several ones.

A normal business domain becomes potential VE partner when it registers the business processes that can offer to a third party VBE. In that case, the domain specifies which local processes can be provided to other domains and under which terms and conditions these processes will be provided, e.g. price.

In general, a process has a name, a set of input parameters, a set of output parameters, a set of sub-processes, a set of tasks, and a set of conditions. The input parameters are the input values to the process, while the output parameters are the output values of the process. A process might consist of sub-processes in a recursive manner. For every sub process, in a similar manner like the process, a name, input and output parameters, sub-processes, sub-tasks and conditions can be specified. This leads to a directed acyclic graph of processes, sub processes and tasks. A task is considered the final, unique, elementary piece of activity that can be included within a process. Actually, tasks are the computational elements of the process while processes and sub-processes orchestrating and coordinating the scenario of the process by scheduling the tasks based on conditions. With every process, sub-process, and task conditions can be associated. Conditions are logical expressions related to input, output, and external values with some logical operators. When a condition, which is related to a process or task, is true, then the associated process or task should be scheduled. By this decomposition of processes into sub-processes and tasks a complex service can easily be described.

When a process consists of sub-processes and tasks belonging to the same administrative domain and can be provided in an autonomous way by this domain, then the process is called local. If there is one sub-process that cannot be provided by this domain, then this sub-process is called remote and, in that case, a supplier domain should be found. If, for this remote process, one static supplier has been found, then the VE relationship is called static. If, for every remote process, a supplier is found dynamically during the process provision, then the VE relationship is called dynamic. If the partners of a VE have not static relationships among them, but on the contrary, they negotiate among each other during process execution then the VE is called dynamic VE.

From the above description and definitions, it is obvious that different administrative domains participate in the execution and management of dynamic VE services. These domains are the:

- **Customer domain**: this is the domain of the user that deploys the services of the VE. The user in this domain can start a service, or terminate it. When the service is completed the results of the service are returned to the customer. Additionally, if, during the execution of the service, a critical situation occurs, the service is aborted and the customer is informed about the event.
- **VE Representative domain**: this is actually the domain that represents the VE to the external world. It executes and manages processes in a transparent to the customer way by deploying the capabilities of the VBE and the remote processes of other business domains. The VE representative domain provides and manages the execution of the VE services by conducting the VBE, locating candidate partners, negotiating with them, and selecting the best one for the execution of the remote processes,
- **VE Candidate/Partner domain**: this is the domain that offers a set of business processes to the VBE and registers certain offers related to specific service templates for potential co-operation with other domains. If this domain is finally selected after a negotiation process it becomes the VE partner domain that will provide the agreed processes to other domains,
- **VE Breeding Environment domain**: this is a third party domain that provides the service templates that the VE candidate partners use to register their offers. This domain manages the offers registered by the VE candidate domains, and provides retrieval services for the selection of VE candidate partners. This domain does not actively participate in the VE and thus does not provide any type of business process management services.

The dynamic VE is meant to establish a dynamic organization through the synergetic combination of dissimilar companies with different core competencies, thereby forming a best of everything consortium to perform a given business project to achieve maximum degree of customer satisfaction. VE is a temporary consortium of partners from different organizations established to fulfill a value adding task, for example fulfilling a product or service demand of a customer. The lifetime of a VE is typically restricted: it is created within the VBE for a definite project and dissolved after the project has been completed.

There are many technical advances that need to be made before the VE lifecycle can be fully automated. These can be organized according to the VE lifecycle, which is composed of creation, operation and dissolution as shown in Figure 4.2.

In our case, we focused primarily on using dynamic virtual enterprises to satisfy customer requirements for non-standard products and services. Hence, any market opportunity placed by a customer requires domain expertise to create an appropriate team of registered members to work on the project. This domain expertise is provided by the VE coordinator either through in-house staff or through registered domain experts.

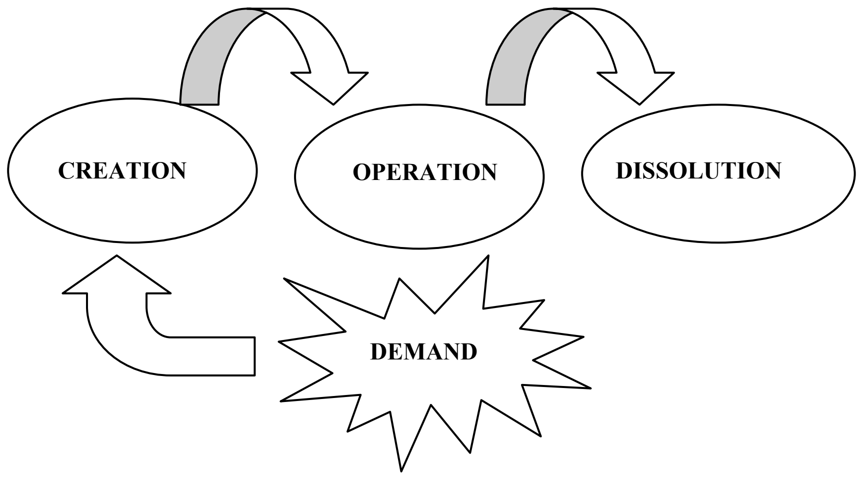


Figure 4.2 VE life cycle

The operation of the dynamic virtual enterprise is no different than any other project operation. To improve productivity, the project members need access to appropriate tools for collaboration as well as tools for project management, etc. These collaboration tools can be provided by the VE coordinator or by external Application Service Providers (ASPs).

The dissolution of the dynamic virtual enterprise can result in either the project team disbanding and moving on to other opportunities or in some cases registering the dynamic virtual enterprise as a vendor organization and competing for other market opportunities. The dissolution step is also a checkpoint for collecting appropriate performance measures for the project and providing feedback on improving the operation of future dynamic virtual enterprises as well as individual companies within the project.

CHAPTER 5

ICT REFERENCE ARCHITECTURE & STRUCTURED METHODOLOGY

The aim of this chapter is to address the second research question: *What framework is needed to structure the body of knowledge related to preparing, setting up and operating virtual enterprises?*

5.1 ICT Reference Architecture

Architecture is the design and contents of an organized environment for running a collection of objects. A wider definition would include within its scope the design of the total built environment, from the macro level to the micro level. Reference architecture is a resource containing a consistent set of architectural best practices for use by all the teams in your organization.

The aim of the ICT reference architecture in this study is to establish and manage virtual enterprise systems with an n-tier client/server technology of Microsoft .NET framework. This reference architecture attempts to portray in a diagrammatic fashion how different enterprises may exchange and use information between their respective organizations' specific proprietary systems and a central server. The central server maintains and enables access to common data sources (released information and documents) and services that are used within virtual enterprise systems. The conceptual scheme of the architecture is given in Figure 5.1. A description of each layer follows.

5.1.1 VE server

This constitutes the "central" VE server domain. It contains three distinct layers:

- **Storage:** this layer is the one which describes databases, backups, etc. In short, this is the VE server repository. This layer can also be described as the persistence layer. A document management system would, for example, fall in this layer.
- **Service:** this layer describes the services which are made available through the VE server. Typically these services will operate over the data/information contained in the storage layer. This layer may also be described as the *business* logic layer, as it contains the rules and definitions to carry out certain tasks on the content held within the storage layer. Some simple examples would include management of shared information, calendaring, access to documents, etc.
- **Access:** this layer describes the different forms and mechanisms available for access control to the VE server. Once access is granted, a user would be able to access the services available in the service layer. Furthermore, this layer in conjunction with the service layer could offer *audit* trailing capabilities. While primarily concerned with access control, this layer could also present some interaction interfaces to the outside. It is to be noted that the end-user or external applications would interact with the *VE Server* through this layer. This layer can also be described as a *security layer*. Some examples could include https, simple access control using some scripts written in aspx, asp, jsp, etc.

5.1.2 Network standards

This constitutes the path and mechanism through which organization specific applications/environments of an end user would interact with the VE Server. It consists of one distinct layer:

- **Communication:** this layer describes the various protocols, methods, and formats through which an end-user or application could potentially communicate with the *VE Server*. In simple terms, this layer acts as the *interface* through which the *Enterprise Systems* and *VE Server* communicate. Relevant examples include internet, TCP/IP, etc.

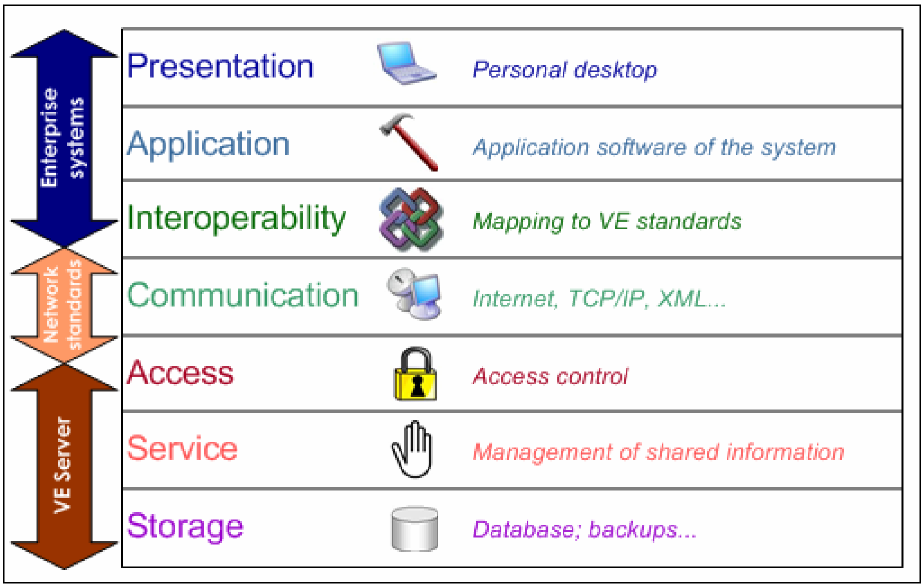


Figure 5.1 Layered ICT reference architecture

5.1.3 Enterprise systems

This constitutes the working environment and systems of the end-user through his/her organization specific applications and platforms. It consists of three layers:

- **Application:** this layer describes different organization specific applications. It may be seen as a composite of one or more of the persistence, business logic, and/or access layers for a particular application. Some examples for this layer could include the organizational document management system, email server, ERP system, etc.
- **Presentation:** this layer describes the different interfaces and mechanisms through which a user can physically interact with an application. It is to be noted that while some applications would provide such an interface, for others one may need to be built in the form of for example some simple web pages.
- **Interoperability:** this layer describes the mechanisms and standards through which application specific information is *restructured* or *mapped* to comply with mutually agreed upon *standards* for data and information exchange. Since many applications have different internal data/information semantics and representation

norms, this layer is required to ensure and foster compatibility with other applications/services. Typically, these other applications would read in this information and then *re-map* it to the *application specific* data/information semantics and representation norms. Examples for this layer could include: IFC file conversions, conformance to some particular schema, etc.

It should be noted that certain elements may span across more than one layer.

5.2 Structured Methodology

The proposed ICT reference architecture which has been built up on n-tier client/server technology, inherits very complex and advanced computer technologies such as database systems, and web based interface. The design and implementation of such a system to virtual enterprise domain needs a structural approach for design and implementation. Otherwise development of a successful system will not be possible.

The structured approach for design and implementation of the virtual enterprise systems with proposed ICT reference architecture is depicted in Figure 5.2. This structured methodology has three phases. A combination of techniques is used. Existing methodologies are not sufficient and sometimes too complex to use. It is advocated here that a combination of techniques/methodologies may be used for the definition of inter-enterprise interfaces. Therefore, one such combination of IDEF0, IDEF1X and UML is presented.

The first phase is the analysis phase where the implementation domain is analyzed and knowledge is extracted to be used for the design phase. The identified, low-level processes are translated and represented as a sequential flow of functions using IDEF0 conventions. Each low-level process is broken down into the form of one or more documented use cases. This is an iterative process until all required use cases have been defined and redundant ones removed. A use case provides a functional description of a system and its major processes and places a boundary on the problem to be solved. It also provides a graphical description of who will use the system and what kinds of interactions they can expect to have within the system. The scenarios in the system are modeled using the activity diagrams. Flows of activities in some use cases are modeled to clarify these use cases. Activity diagrams are used to model the flow of activities in a procedure. They provide very broad views of business processes or they can be used to break out the activities that occur within a use case.

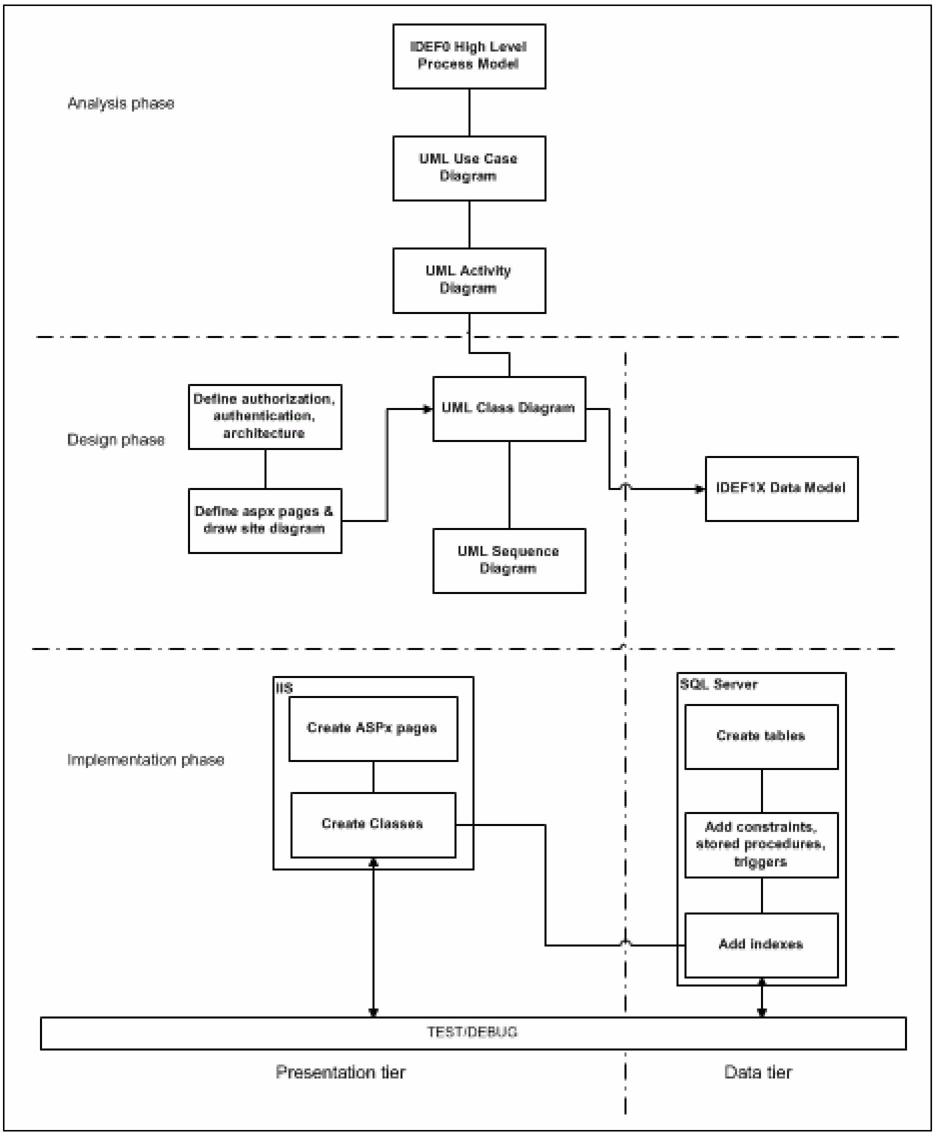


Figure 5.2 Structured methodology for virtual enterprise systems

In the design phase, for all tiers of the system a detailed design activity is carried on.

The details and specifics of this design activity are highly dependent on the features of the client/server technology of Microsoft .NET. Using the textual description of the use cases, a noun-phrase analysis is conducted. The main objective is to identify the main actors (nouns) and processes (phrases) that are required to deliver the functionality of the system. Removal of redundancy, grouping of similar items etc. lead to simplistic identification of the main system modules (classes). Using the identified system actors and processes, the use cases are updated and sequence diagrams (at least one per use case) generated. These typically describe the different interactions between the classes and the messages they exchange with each other. One of the objectives of design phase is to localize data storage by distributing the information that increases the robustness of the system. This step leads mainly to database modeling using IDEF1X technique to define entities and their relationships according to system modules in class diagram. Generally, this phase covers the design of the data structure, design of the class structure and design of the Internet based interface of the system.

The last phase is the implementation phase. This phase covers the implementation of the designed system using the technologies of Microsoft .Net solution. Several tools are used for the development and hosting of the developed components. During the implementation phase, the code is written, database tables are created and web pages are constructed. Then the initial prototype is tested. After satisfactory testing, first iteration in system development ends and the second iteration start with the analysis phase again. This goes on like that until the system works seamlessly. The structured methodology is explained in detail in the following sections.

5.2.1 Analysis phase

5.2.1.1 IDEF0 model

A model is a representation of a set of components of a system or subject area. The model is developed for understanding, analysis, improvement or replacement of the system. Systems are composed of interfacing or interdependent parts that work together to perform a useful function. System parts can be any combination of things, including people, information, software, processes, equipment, products or raw materials. The model describes what the system does, what controls it, what things it works on, what means it uses to perform its functions and what it produces.

IDEF0 is a modeling technique based on combined graphics and text that are presented in an organized and systematic way to gain understanding, support analysis, provide logic for potential changes, specify requirements, or support systems level design and integration activities. An IDEF0 model is composed of a hierarchical series of diagrams that gradually display increasing levels of detail describing functions and their interfacing within the context of the system. There are three types of diagrams: graphical, text, and glossary. The graphical diagrams define functions and functional relationships via box and arrow syntax and semantics. The text and glossary diagrams provide additional information in support of graphic diagrams. For the complete syntax and semantics and development procedures refer to Draft Federal Information Processing Standards Publication 183 for IDEF0 (1993).

IDEF0 models provide a "blueprint" of functions and their interfaces that must be captured and understood in order develop a solution on an existing system. The IDEF0 model reflects how system functions interrelate and operate just as the blueprint of product reflects how the different pieces of a product fit together. In summary, application of an IDEF0 model in the analysis phase will provide the following benefits before proceeding into design details:

- *Activity Modeling Graphic Representation:* The box and arrow graphics of an IDEF0 diagram show the business process as a box. In order to be able to express real-life manufacturing operations, boxes may be interpreted as operating with other boxes, with the interface arrows providing "constraints" as to when and how operations are triggered and controlled.
- *Conciseness:* The documentation of a VE architecture must be concise to permit encompassing the subject matter. The linear, verbose characteristic of ordinary

language text is insufficient. The two-dimensional form provided by a blue-print like language has the desired conciseness without losing the ability to express relationships such as interfaces, feedback and error paths.

- *Communication:* IDEF0 provides a communication language that is understandable among analysts, designers, developers, users and managers.
- *Methodology:* Step-by-step procedures are provided for modeling, review and interview tasks.
- *Organization vs. function:* The separation of organization form function is included in the purpose of the model and carried out by the selection of functions and arrow labels during model development.

5.2.1.2 UML use case diagram

A use case diagram provides a functional description of a system and its major processes and places a boundary on the problem to be solved. It also provides a graphic description of who will use the system and what kinds of interactions they can expect to have with the system. In use case modeling entities outside the problem area that are going to use the application are called "actors". An actor may be a person, another software application, or a hardware device. Processes that occur within the application area are called "use cases". Use case is represented by an oval. The name of the use case is written inside the oval. A rectangle with rounded corners represents the application or the system. A line connecting an actor and a use case represents the interaction between them. Notation of UML use case diagram can be seen in Figure 5.3.

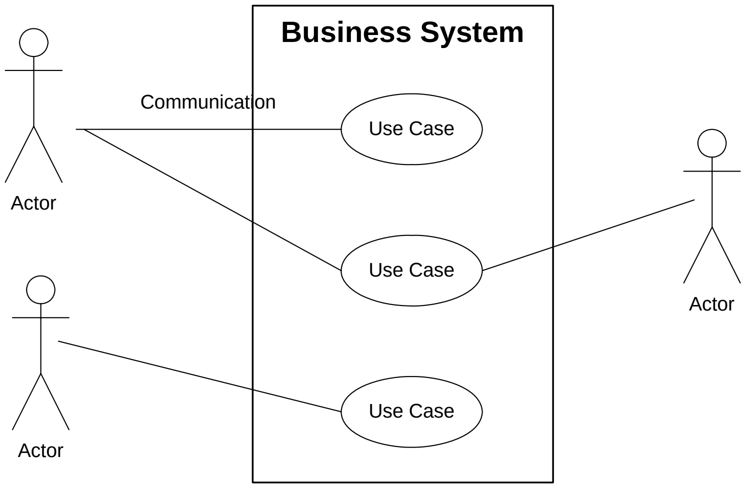


Figure 5.3 Notation for use case diagram

5.2.1.3 UML activity diagram

Activity diagrams are used to model the flow of activities in a procedure. They provide very broad views of business processes or they can be used to break out the activities that occur within a use case. In UML, activities are called states, so activity diagrams are associated with state diagrams. Activity diagrams are just the opposite of the state diagrams. Unlike state diagrams that focus on the events occurring within a single object as it responds to messages, an activity diagram can be used to model an entire business process or to provide a high level view of what is going inside a use case. Events are referred to as transitions. Diamonds are used to allow the developer to show what happens when a transition can have more than one consequence. Bars are used to show when several streams of events must come together before the next activity can occur. State symbols can be used to show the start and finish points. Notation for UML activity diagram can be seen in Figure 5.4.

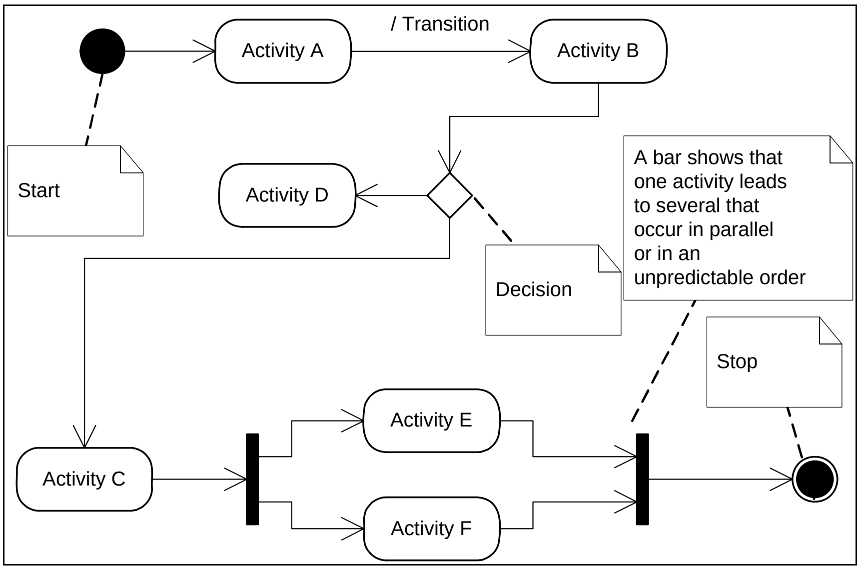


Figure 5.4 Notation for activity diagrams

5.2.2 Design phase

5.2.2.1 Data tier

5.2.2.2.1 IDEF1/X model

In the design phase, the first step is to design the physical structure of the database. This step covers the transformation of entities to SQL tables by adding attributes, primary and foreign keys, resolving many-to-many relations and normalization of data to avoid anomalies. In this approach, IDEF1/X notation is used for the detailed design of the database system. For more detailed information about the notation reader can refer to IDEF1/X, Integration Definition for Information Modeling, Federal Information, Processing Standards Publication 184 (1993).

5.2.2.1.1.1 Attributes

Attributes are characteristics of entities that provide descriptive detail about them. After selecting the suitable entities and their relationships that will take place as tables in the data view of the system from the output of the analysis phase, the attributes of each entity should be added. An example of a table with attributes is depicted in Figure 5.5. Attributes can also be added to relationships. However they are not normally assigned to one-to-one or one-to-many relationships, usually they are assigned to binary many-to-many or ternary relationships. It should be noted that this kind of relationships should also be appeared as tables, and resolution of many-to-many relation will be described in the following sections.

Employee			
Employee_no	Employee_name	Birth_date	Sex
…..			

Figure 5.5 Sample table with attributes

5.2.2.1.1.2 Keys

A key is a set of columns that define a table. There are four types of keys: candidate keys, primary keys, alternate keys, and foreign keys. Along with these types, there are two basic key classes: composite and singleton. A composite key is composed of more than one attribute. A singleton key, on the other hand, is any key made up only one attribute.

Candidate Keys

A candidate key is any possible unique identifier for a row within a database table. A candidate key can be either composite or singleton key. Every table must have at least one candidate key, but it is also possible for a table to have more than one candidate key. The collection of all candidate keys is often referred as super keys.

Primary Keys

A primary key is one or more attributes that uniquely identify a row in a table. One candidate key can become the primary key of the table. Primary key is often depicted as an underline, below the attribute name.

Alternate Keys

Alternate keys are the candidate keys that do not become the primary key.

Foreign Keys

A foreign key exists when one or more attributes in an entity refer to a primary key in another entity. Foreign key is the key point for defining relationships between tables. When there is a relationship between two entities, this means that this relation information should be kept in the physical design of the database. This is achieved by migration of the primary key of one table to the other one in the relation as the foreign key. Three types of relationships are commonly formed by the way foreign keys migrate:

- *Identifying relationship:* An identifying relationship is a mandatory relationship, in which the first table's key columns are migrated as the part of second table's primary key. This relation is also called parent-child relation. This relation guarantees that an instance can not occur in the child table unless there is a related instance in the parent table (Figure 5.6).

- *Non-identifying relationship:* This type of relationship is formed by migration of first table's primary keys to second table as not being a part of its primary key. This type can be divided into sub-types as mandatory and optional. An optional non-identifying relationship allows the non-existence of the related instance at the table where primary key migrates from, whereas mandatory non-identifying relationship does not (Figure 5.7).

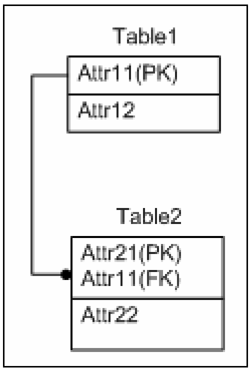


Figure 5.6 Identifying relationship

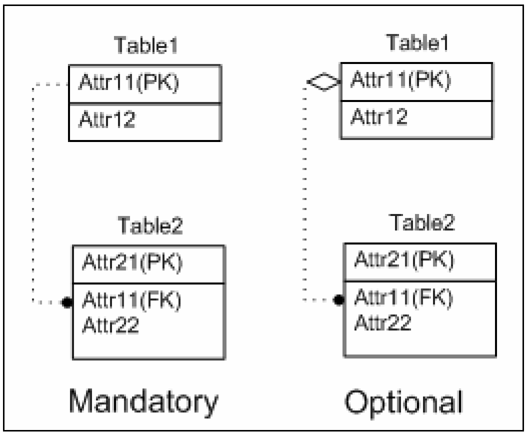


Figure 5.7 Non-identifying relationship

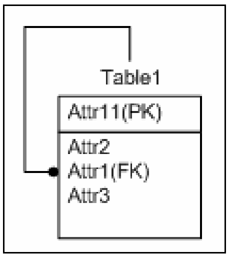


Figure 5.8 Recursive relationship

- *Self-recursive relationship:* This is a table, which is related to itself. The primary key migrates as a foreign key to the same table (Figure 5.8).

5.2.2.1.1.3 Resolution of many-to-many relationship

Many-to-many relations exist between a pair of tables, if a single record in the first table can be related to one or more records in the second table and a single record in the second table can be related to one or more records in the first table. In order to prevent the existence of redundant data a third table should be added, between the first and second table and the relation information should be stored by migrating the primary keys of both tables to this table. The process of resolving many-to-many relation is depicted in Figure 5.9.

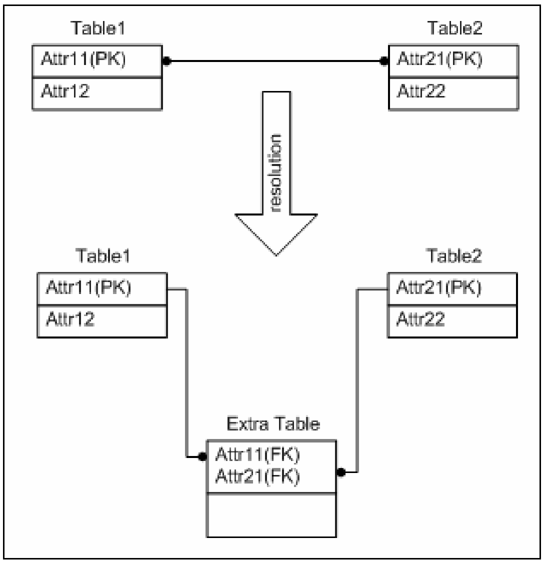


Figure 5.9 Resolving m-m relationships

5.2.2.1.1.4 Normalization

In a database scheme several inconsistency conditions may occur due to the redundancy of data. Redundancy can lead to:

- Redundant storage: Some information is stored repeatedly and storage space is lost.

- Update anomalies: If one copy of repeated data is updated, an inconsistency is created unless all copies are similarly updated.
- Insertion anomalies: It may not be possible to store some information unless some other information is stored as well.
- Deletion anomalies: It may not be possible to delete some information without losing some other information as well.

These are very serious problems and affect the consistency and performance a database scheme directly. In order to avoid these problems, a process called Normalization should be carried out on the scheme of the database. Normalization is the process of refining tables, keys, columns and relationships to create a consistent database design. Several levels of normal forms are proposed. 1NF, 2NF, 3NF are commonly used today in database design. In database design the goal is to set database scheme to 3NF (Benon-Davies, 1991)

In order to explain the normal forms, the basic theory of Functional Dependency should be mentioned here. The property of one or more attributes that uniquely determines the value of one or more attributes is called functional dependency (FD). Let R be a relation schema, and X and Y be non-empty sets of attributes in R. It can be said that an instance r of R satisfies the FD $X \rightarrow Y$ if the following holds for every pair of tuples t1 and t2 in r:

If t1.X = t2.X then t1.Y = t2.Y (Ramakrishnan, 2001)

First Normal Form

Before giving the definition the first normal form, it is better to give the difference between a domain, an attribute, and a column. A domain is the set of all possible values for a particular type of attribute but may be used for more than one attribute. For example, the domain of people’s names is the underlying set of all possible names that could be used for either customer-name or operator-name in a table. Each column in a relational table represents a single attribute, but in some cases more than one column may refer to the same attribute. When this occurs the table is said to have a repeating group and is therefore unnormalized.

A table is in first normal form (1NF) if and only if all columns contain only atomic values; that is, there are no repeating groups (columns) within a row. The table in Figure 5.10 is a table, which is in 1NF.

Pro.Log

WS_no	Type	Dept_name	Dept_addr	Op_ID	Op_name	Op_tel
4512	Mill	Assembly	Fact1	56	john	33445
4512	Mill	Assembly	Fact1	44	eric	23656
5545	Mill	Assembly	Fact1	55	michael	89765
1125	Lathe	ChipM/C	Fact2	53	smith	67896
1125	Lathe	ChipM/C	Fact2	44	eric	23656
1125	Lathe	ChipM/C	Fact2	11	derek	12334

Figure 5.10 Table in first normal form

Second Normal Form

A table is in second normal form (2NF) if and only if it is in 1NF and every non-key attribute is fully dependent on the primary key. An attribute is fully dependent on the primary key if it is only on the right-hand side of FDs for which the left side is either the primary key itself or something that can be derived from the primary key using the transitivity of FDs.

In the example depicted in Figure 5.11, the following international dependencies can be deducted.

Production_Log:

WS_no $\rightarrow$ Type, Dept_name

Op_ID $\rightarrow$ Op_name, Op_tel

Ws_no $\rightarrow$ Dept_addr (Transitive dependency as WS_no $\rightarrow$ Dept_name and Dept_name $\rightarrow$ Dept_addr)

The WS_no and Op_id is the only candidate key and therefore is the primary key. None of the FDs contain whole of the primary key on the left-hand side of the arrow so, the table does not satisfy the condition for 2NF for any of the FDs. However, after the table, which is in Figure 5.10 is separated into different tables as in Figure 5.11, the FDs for each table that satisfies the criteria of 2NF and as follows:

WS:

WS_no $\rightarrow$ Type, Dept_name

WS_no $\rightarrow$ Dept_addr (Transitive dependency as WS_no $\rightarrow$ Dept_name and Dept_name $\rightarrow$ Dept_addr)

Operator:

Op_ID $\rightarrow$ Op_name, Op_tel

WS_Op:

WS_no, Op_ID is a candidate key (no FDs)

Third Normal Form

As 1NF and 2NF eliminates most of the redundant data, they are still not enough because of occurrence of anomalies caused by the transitive dependencies. A table is in third normal form when all non-key attributes are not functionally dependent on each other. In other words, a table can be in 3NF, when all the non-key attributes are dependent on the primary key, nothing but the primary key. It can be observed in Figure 5.12 that the transitive dependency is eliminated by dividing WS table into two tables as WS and Dept and all of the tables preserve the conditions of 3NF.

WS

WS_no	Type	Dept_name	Dept_addr
4512	Mill	Assembly	Fact1
4512	Mill	Assembly	Fact1
5545	Mill	Assembly	Fact1
1125	Lathe	ChipM/C	Fact2
1125	Lathe	ChipM/C	Fact2
1125	Lathe	ChipM/C	Fact2

Operator

Op_ID	Op_name	Op_tel
56	john	33445
44	eric	23656
55	michael	89765
53	smith	67896
44	eric	23656
11	derek	12334

WS_Op

WS_no	Op_ID
4512	56
4512	44
5545	55
1125	53
1125	44
1125	11

Figure 5.11 Tables in 2NF

WS

WS_no	Type	Dept_name
4512	Mill	Assembly
4512	Mill	Assembly
5545	Mill	Assembly
1125	Lathe	ChipM/C
1125	Lathe	ChipM/C
1125	Lathe	ChipM/C

Operator

Op_ID	Op_name	Op_tel
56	john	33445
44	eric	23656
55	michael	89765
53	smith	67896
44	eric	23656
11	derek	12334

WS_Op

WS_no	Op_ID
12	56
4512	44
5545	55
1125	53
1125	44
1125	11

Dept

Dept_name	Dept_addr
Assembly	Fact1
Assembly	Fact1
Assembly	Fact1
ChipM/C	Fact2
ChipM/C	Fact2
ChipM/C	Fact2

Figure 5.12 Tables in 3NF

5.2.2.2 Presentation tier

For the development of the presentation tier there are two main ways. The form based Graphical Using Interfaces (GUI) as native Win32 executables or WWW based interface. The right decision is WWW based interface for several reasons:

- Accessible from every client on the network or internet
- No work for client machine loading or maintenance
- Continuous support and addition of new features, standards and technologies by software industry.

When it is not a simple WWW interface for a presentation site but a data-oriented, business or control application it needs scripting for the WWW interface. There are two choices: client or server side scripting. They both have advantages and disadvantages, the selection should be made depending on the case and also it should be kept in mind that mixed is also possible.

5.2.2.2.1 Client-side scripting

VB.NET is a powerful object-oriented language and one of the most powerful tools when developing internet based systems. It can be the glue that ties everything together within an HTML document; HTML forms, ActiveX controls, Java applets, and the Web browser itself. Client-side scripts are actually small programs that are embedded in HTML text. When the HTML page is transferred to the client, they are also transferred, and the client browser can compile and execute them instantly. Because these scripts run entirely in the client's browser, they are called client-side scripts.

Microsoft Internet Explorer can execute client-side scripts that are written in Jscript, Microsoft's open implementation of Netscape's JavaScript language or VB.NET. The most important deficiency of client-side scripting is that if the browser does not support the script language you use it does not work. The advantages and disadvantages of client side scripting can be summarized as follows:

Advantages:

- Fewer loads on server. Data validation can be performed on client machine.
- Allows for greater interactivity with the Web page in-place.
- Does not require additional download time for round trip to the server.

Disadvantages:

- Code is exposed to the user.
- Relies on the browser's support of both the script engine and the browser object model.

5.2.2.2.2 Server-side scripting

Server side scripting allows the developer to use tools that reside on the server. The famous method of Microsoft's server side scripting is called ASP.NET. ASP.NET pages are actually text pages like HTML pages. The main difference is that they have script code embedded in them. The script either again written by Jscript or VB.NET, but this time they are executed in the server and only raw HTML output is send to the client. Server-side scripting allows unleashing the power of components that reside on server. Advertisements, database access, counters, file access are examples of server-side components that are accessible through ASP.NET. Internet Information Server (IIS) 5.0 as well as Personal Web Server XP support ASP.NET. The advantages and disadvantages of server-side scripting can be summarized as follows:

Advantages:

- Leverage the robust resources of a central, high-powered machine.
- Create a browser-independent site because all script code is executed on the web server machine.
- Code is not exposed to user.

Disadvantages:

- Relies on a web server that supports server-side scripting.
- Requires resources on the server to be available for multiple users simultaneously.

5.2.2.2.3 Authentication

Internet Information Server can provide three different security tests for users.

- Anonymous user: All clients can access to pages with anonymous account that IIS placed in XP users which is namely IUSR_machine name.
- Basic Authentication (Clear Text): Identification information is transferred as clear text from client to server. There is always a risk of subject to packet sniffing, which is very easy to do on the network.
- Integrated Windows: Also known as NTLM authentication. This is very secure as the identification is encrypted but it has two limitations.
    - The browser should be NTLM compatible like Internet Explorer 2.x and higher.
    - The connection between the browser and the server must allow RPC calls. Although this is usually the case in intranets, most proxy servers and firewalls are configured to block RPC ports for security issues.

5.2.2.2.4 Authorization

There are two basic approaches to authorization:

- Role based. With a role-based (or operations-based) approach to security, access to operations (not back-end resources) is authorized based on the role membership of the caller. Roles (analyzed and defined at application design time) are used as logical containers that group together users who share the same security privileges (or capabilities) within the application. Users are mapped to roles within the application and role membership is used to control access to specific operations (methods) exposed by the application.

Where within your application this role mapping occurs is a key design criterion; for example:

- On one extreme, role mapping might be performed within a back-end resource manager such as a database. This requires the original caller's security context to flow through your application's tiers to the back-end database.
- On the other extreme, role mapping might be performed within your front-end Web application. With this approach, downstream resource managers are accessed using fixed identities that each resource manager authorizes and is willing to trust.
- A third option is to perform role mapping somewhere in between the front-end and back-end tiers; for example, within a middle tier Enterprise Services application.

In multi-tiered Web applications, the use of trusted identities to access back-end resource managers provides greater opportunities for application scalability (thanks to connection pooling). Also, the use of trusted identities alleviates the need to flow the original caller's security context at the operating system level, something that can be difficult (if not impossible in certain scenarios) to achieve.

- Resource based. The resource-based approach to authorization relies on Windows ACLs and the underlying access control mechanics of the operating system. The application impersonates the caller and leaves it to the operating system in conjunction with specific resource managers (the file system, databases, and so on) to perform access checks.

  This approach tends to work best for applications that provide access to resources that can be individually secured with Windows ACLs, such as files. An example would be an FTP application or a simple data driven Web application. The approach starts to break down where the requested resource consists of data that needs to be obtained and consolidated from a number of different sources; for example, multiple databases, database tables, external applications or Web services.

The resource-based approach also relies on the original caller's security context flowing through the application to the back-end resource managers. This can require complex configuration and significantly reduces the ability of a multi-tiered application to scale to large numbers of users, because it prevents the efficient use of pooling (for example, database connection pooling) within the application's middle tier.

5.2.2.2.5 Site diagram

The creation of good Web interface requires considerable effort. In order to ease this task, it is a good approach to draw a hierarchical site diagram, which consists of ASPx and HTML pages and the hierarchical relations among them. This site diagram will then be transferred to a real structure using the tools of Visual Studio .NET during implementation stage. A sample site diagram is given in Figure 5.13.

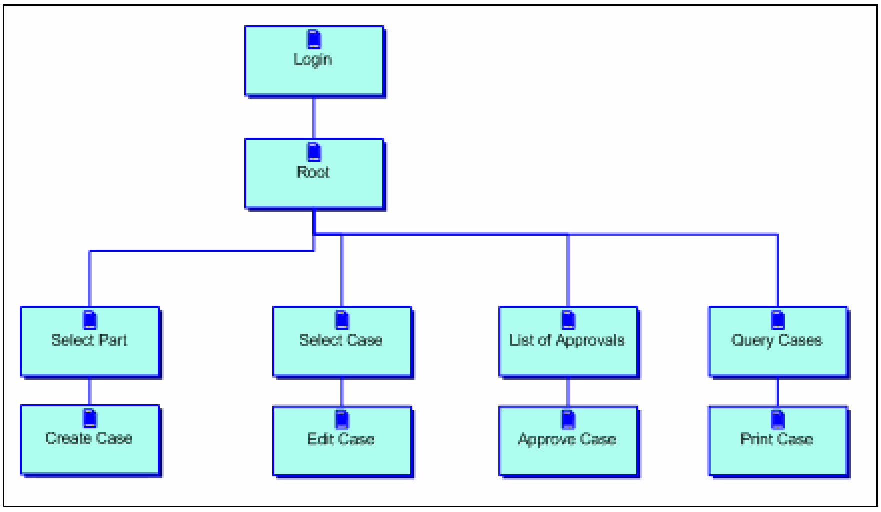


Figure 5.13 Sample site diagram design

5.2.2.2.6 UML class diagram

The UML class diagram shows classes and the relationships between classes. The simplest way of representing a class in UML is to draw a rectangle and place the name of the class inside the rectangle. Later this class can be detailed by dividing the rectangle into three horizontal compartments. In the top section the name of the class is indicated, in the second section the attributes are listed, and in the lower section the methods are listed. Notation for classes can be seen in Figure 5.14.

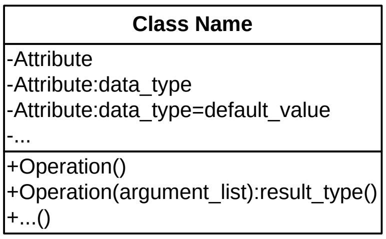


Figure 5.14 Notation for classes

Any physical or conceptual connection between two or more classes is documented by “association”. The most generic association is represented with a line between two classes. An association is naturally bidirectional. In other words, an association implies that the objects of one class will send messages to operations associated with the objects of the second class and that the second class’s objects will respond when appropriate. The name of the association is written just above or below the line connecting the two classes. But, if the association represents a single obvious relationship, its name can be omitted. The name of the role played by the class is called “role”, and it is written just over or below the association line, next to the class. Notation for association can be seen in Figure 5.15.

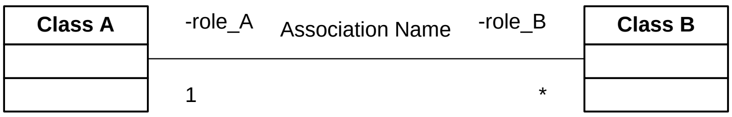


Figure 5.15 Notation for an association

UML also provides a way to indicate the multiplicity of an association. “Multiplicity” refers to how many objects of one class can relate to each object of another class. Notation for multiplicity of associations can be seen in Figure 5.16.

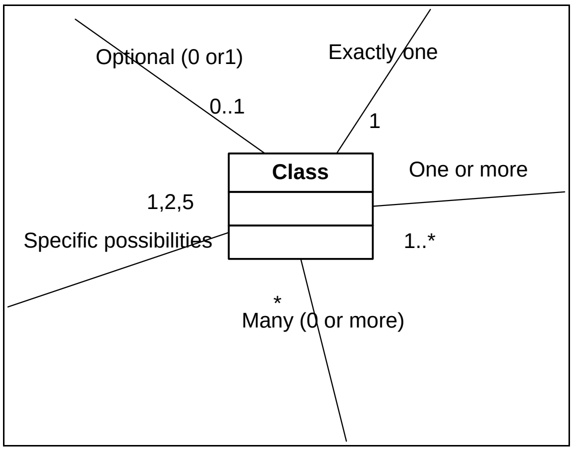


Figure 5.16 Notation for multiplicity of association

“Generalization” refers to a relationship when one class shares its properties, methods, relationships, and constraints with another class. UML indicates generalization by means of hollow-headed arrow. The arrow always points towards the super-class. Notation for generalization can be seen in Figure 5.17.

An “aggregation” refers to part-whole relationships. It is indicated by placing a small diamond at the end of the association line that runs between part classes and the whole class, pointing at the whole class. Notation for aggregation can be seen in Figure 5.18.

Composition relationships are strong form of aggregation. It is indicated by placing a filled diamond at the end of the association line. Composition indicates that the lifetime of Class B and Class C is dependent upon Class A Notation for composition can be seen in Figure 5.19.

“Constraints” are functional relationships between elements of a class or object model. A constraint sets limits on the values that one of the elements can take. UML allows write notes on diagrams by putting them in rectangles with a dog-eared right corner. Notation for constraints and notes can be seen in Figure 5.20.

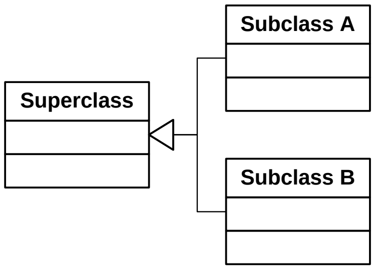


Figure 5.17 Notation for generalization

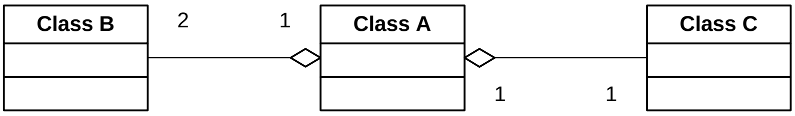


Figure 5.18 Notation for aggregation

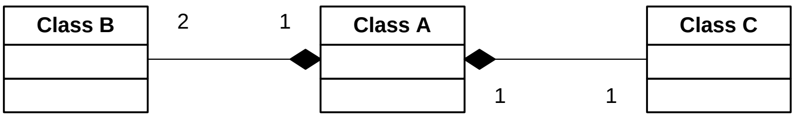


Figure 5.19 Notation for composition

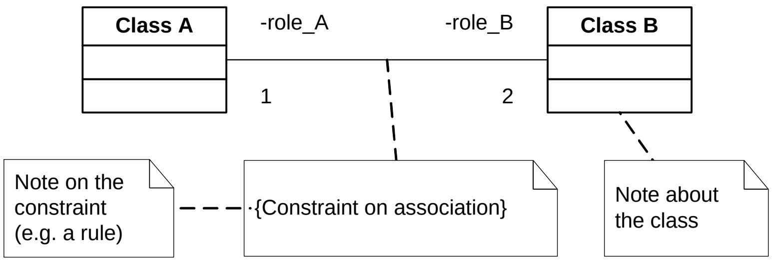


Figure 5.20 Notation for a constraint and for notes

In some situations, objects of classes may need to be diagramed. UML indicates an object as a rectangle like a class. However, the difference between an object and a class can be noticed by the way the name is written inside the rectangle. An object can be named in UML in three ways:

- :Class Name
- object name:
- object name:Class Name

Notation for objects can be seen in Figure 5.21.

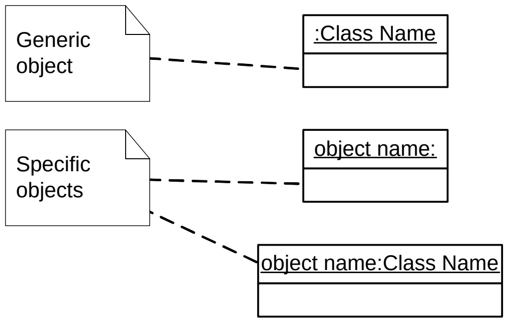


Figure 5.21 Notation for objects

5.2.2.2.7 UML sequence diagram

A sequence diagram is a graphical way to illustrate a scenario. They show the sequence in which objects pass messages to each other. They can also show how long an object takes to process a message, when objects are created and destroyed, and when objects are synchronous and asynchronous. To create sequence diagram, a set of classes that will be involved in a scenario are identified. Then the objects of these classes are listed in a line along the top of a page. The dotted lines dropped beneath each object represent the lifelines of the objects. The line on the left often represents an actor or an event outside the system boundary. An event arrow connects objects. In effect, the event arrow suggests that a message is moving between these two objects. Events are given names, which can be a phrase or an operation or message name. Constraints are listed within braces. The rectangles that lie below the object rectangles and over the dotted lines indicate when the object is active. The cross symbol at the end of the dotted line beneath an object tells that object is destroyed. Notation for UML sequence diagram can be seen in Figure 5.22.

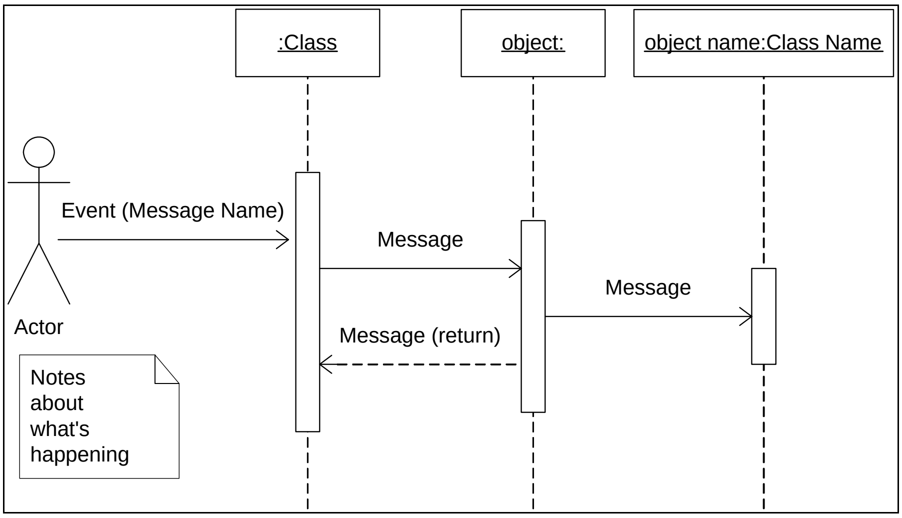


Figure 5.22 Notation for a sequence diagram

5.2.3 Implementation phase

5.2.3.1 Data tier

Depending on the Universal Data Access paradigm, theoretically it is possible to access every kind of data resources like structured data sources, like, MS-Access, MS-Exchange Server, FoxPro, Oracle, Legacy systems or unstructured data like files. However SQL Server as a high performance Internet database server is highly recommended and used all through this study. Therefore the key points, techniques and recommendations will be given about using this product in this section.

5.2.3.1.1 Components of SQL server

The SQL Server services include MSSQL Server, SQL Server Agent, MS Distributed Transaction Coordinator and MS Search:

- *MSSQLServer service* is the database engine. It is the component that processes all Transact-SQL statements and manages all files that compromise the databases on the server.
- *SQLServer Agent service* is a service that works in conjunction with SQL Server to create and manage local or multi-server jobs, alerts, and operator.
- *MS Distributed Transaction Coordinator service* allows clients to include several different sources of data in one transaction. MS DTC coordinates the proper completion of distributed transactions to ensure that all updates on all servers are permanent or in case of errors, that all modifications are cancelled.
- *Microsoft Search service* is a full-text engine that runs as a Windows XP service. Full-text support involves the ability to issue queries against character data and the creation and maintenance of the indexes that facilitate these queries.

SQL Server 2000 Enterprise manager is a design and implementation environment where most of the activities can be realized with graphical user interface and wizards (Figure 5.23). However all of the activities depend on the T-SQL scripts and in following sections, the scripts samples for implementation specifics will be given.

5.2.3.1.2 Creating a database

When creating a database, it is important to determine the name of the database, its owner (the user who creates the database), its size, and the files and filegroups used to store it. The following facts are important when creating a database:

- Permission to create a database defaults to members of the **sysadmin** and **dbcreator** fixed server roles, although permissions can be granted to other users.
- The user who creates the database becomes the owner of the database.
- A maximum of 32,767 databases can be created on a server.
- The name of the database must follow the rules for identifiers.

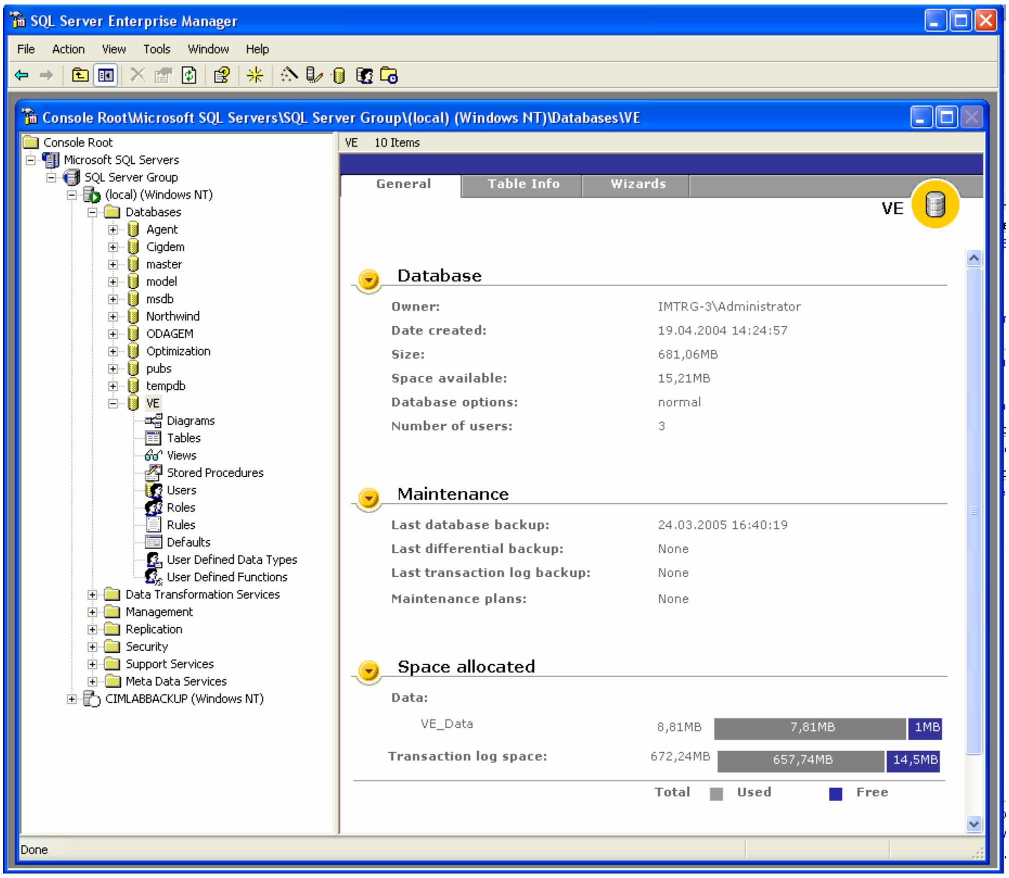


Figure 5.23 SQL server enterprise manager

Three types of files are used to store a database:

- Primary files: These files contain the startup information for the database. The primary files are also used to store data. Every database has one primary file.
- Secondary files: These files hold all the data that does not fit in the primary data file. Databases do not need secondary data files if the primary file is large enough to hold all the data in the database. Some databases may be large enough to need multiple secondary data files, or they may use secondary files on separate disk drives to spread the data across multiple disks.
- Transaction log: These files hold the log information used to recover the database. There must be at least one transaction log file for each database, although there may be more than one. The minimum size for a log file is 512 kilobytes (KB).

5.2.3.1.3 Creating a table

When creating a table in SQLServer, the table name, column names, and column data types should be specified. Column names should be unique to a specific table, but same name can be used in different tables. The following script creates a table:

```
CREATE TABLE member
    (member no      int         NOT NULL,
    name            char(50)    NULL,
    surname         char(50)    NULL)
```

Consider the following facts about tables in SQL Server:

- Two billion tables per database
- 1024 columns per table
- 8092 bytes per row

5.2.3.1.4 Enforcing data integrity with SQL server

Data integrity refers to consistency and accuracy of data that is stored in a database. Four different types of data integrity exist:

- Entity integrity defines a row as a unique entity for a particular table. Entity integrity enforces the integrity of the identifier column(s) or the primary key of a table (through indexes, UNIQUE constraints, PRIMARY KEY constraints, or IDENTITY properties).
- Domain integrity is the validity of entries for a given column. Domain integrity can be enforced by restricting the type (through data types), the format (through CHECK constraints and rules), or the range of possible values (through FOREIGN KEY constraints, CHECK constraints, DEFAULT definitions, NOT NULL definitions, and rules).

- Referential integrity preserves the defined relationships between tables when records are entered or deleted. In Microsoft SQL Server 2000, referential integrity is based on relationships between foreign keys and primary keys or between foreign keys and unique keys (through FOREIGN KEY and CHECK constraints). Referential integrity ensures that key values are consistent across tables. Such consistency requires that there be no references to nonexistent values and that if a key value changes, all references to it change consistently throughout the database. When referential integrity is enforced, SQL Server prevents users from:
    - Adding records to a related table if there is no associated record in the primary table.
    - Changing values in a primary table that result in orphaned records in a related table.
    - Deleting records from a primary table if there are matching related records.
- User-defined integrity allows defining specific business rules that do not fall into one of the other integrity categories. All of the integrity categories support user-defined integrity (all column- and table-level constraints in CREATE TABLE, stored procedures, and triggers).

5.2.3.1.5 Stored procedures

A stored procedure is a named collection of Transact-SQL statements that is stored on the server. They are a method of encapsulating repetitive tasks that executes efficiently. Stored procedures support user-declared variables, conditional execution, and other powerful programming languages. They are similar to procedures in programming languages in the following ways:

- Contain statements that perform operations in a database, including the ability to call other stored procedures.
- Accept input parameters.
- Return a status value to a calling stored procedure or batch to indicate successor

failure.

- Return multiple values to the calling procedures or batch in the form of output parameters.

When a stored procedure is created, the statements in it are parsed for syntactical correctness. SQL Server then stores the name of the stored procedure in the *sysobjects* system table and the text of the stored procedure in the *syscomments* system table in the current database. The first time that stored procedure is executed, the query processor reads the stored procedure in a process called resolution. When the stored procedure is successfully resolved, the SQL Server query optimizer analyzes the Transact-SQL statements in the stored procedure and creates a plan that contains the fastest method to access the data. Lastly the stored procedure is compiled and the execution plan is placed in the procedure cache.

When the same stored procedure is executed subsequently the SQL Server directly uses the optimized execution plan in the procedure cache. This saves a large amount of CPU time and accelerates execution of stored procedures. SQL Server ages old, unused plans out of the cache only when space is needed.

The following script is an example for creating a stored procedure:

```
CREATE PROC retrieve_members
AS
SELECT *
FROM member
WHERE age>30
```

It is also possible to control atomicity of a transaction within a stored procedure using the BEGIN TRANSACTION, COMMIT TRANSACTION and ROLLBACK TRANSACTION phrases.

There are several advantages of using stored procedures:

- Share application logic with other applications or modules, thereby ensuring consistent data access and modification.
- Provide security mechanisms.
- Improve performance, as SQL Server will only execute a single execution plan after the first run.
- Reduce network traffic as clients can execute a stored procedure by just a call instead of sending all the T-SQL statements.

5.2.3.1.6 Triggers

A trigger is a special kind of stored procedure that executes whenever an attempt is made to modify data in a table that the trigger protects. When an attempt is made to insert, update or delete data in a table and a trigger is defined for that particular action has been defined on that table, the trigger executes automatically. Unlike stored procedures, triggers can not be called directly and do not pass or accept parameters.

Triggers are best used to maintain low-level data integrity, not to return query results. The primary benefit of triggers is that they can contain complex processing logic. By using triggers, it is possible to invoke predefined or dynamic custom error messages when certain conditions occur as a trigger executes.

5.2.3.1.7 Views

Views also known as virtual tables provide a useful mechanism for restricting users to certain subsets of data and allowing users to access customized logical aspects of the data. Views usually contain multiple base-table joins and complex aggregations or return large result sets; therefore, without the aid of an index, views frequently suffer from poor performance.

The functionality of SQL Server views is expanded to provide system performance benefits through the use of indexed views. Creating a unique clustered index on a view as well as non-clustered indexes can improve data-access performance on the most complex queries.

An indexed view is any view that has a unique clustered index. At the time a CREATE INDEX statement is executed on a view, the result set for the view is materialized (expanded) and stored in the database with the same structure as a table that has a clustered index. The indexed view automatically reflects modifications made to the data in the base tables after the index is created, the same way an index created on a base table does. The requirement that the view's clustered index be unique improves the efficiency with which SQL Server can find the rows in the index that are affected by any data modification.

When a standard view is created, the meta data (or description of the data) is defined by encapsulating a SELECT statement that defines a result set to be represented as a virtual table. When a view is referenced in the FROM clause of another query, this meta data is retrieved from the system catalog and materialized in place of the reference to the view. After the view is expanded, the query optimizer compiles a single execution plan for the executing query.

5.2.3.1.8 Indexes

There two methods that SQL Server finds any data:

- Scanning all the data pages of tables, called table scan.
- Using indexes, by traversing the index tree structure to find the rows only needed for criteria.

Using indexes may increase the system performance. SQL Server uses three types of index structures:

- Clustered indexes

- Non-clustered indexes
- Non-clustered indexes built on top of clustered indexes.

Performance tuning with indexes is highly advanced process and needs in depth knowledge of index structures and good experience with using them. However index-tuning wizard can be a good solution for inexperienced developers. Index tuning wizard requires a workload to generate a recommended index configuration. A workload is a representative sample of the database activity. It can be created by using SQL Server Profiler. So the recommended practice about indexes is to tune them after the development. When the system started to be used by real users, a workload should be generated. Then by the use of this workload analysis, index tuning wizard will find the optimum index combination by itself.

5.2.3.2 Presentation tier

When creating a site, the organization of the pages and how the users will navigate through these pages, are very important. It is a good point here to start with the site diagram that have been developed in the design stage and transfer this site architecture to the project by using site designer utility of Visual Studio .NET. When a certain theme is used during the creation of the project, the architecture in the site designer will update the navigational links in the pages automatically.

Programming ASP.NET pages heavily based on server-side scripting. For some reason like validation or device control, client-side scripting can also be added. Design features of Visual Studio .NET enables mixed scripting and provides the power of many tools like HTML controls, web or design time controls, and many others in one common design environment (Figure 5.24).

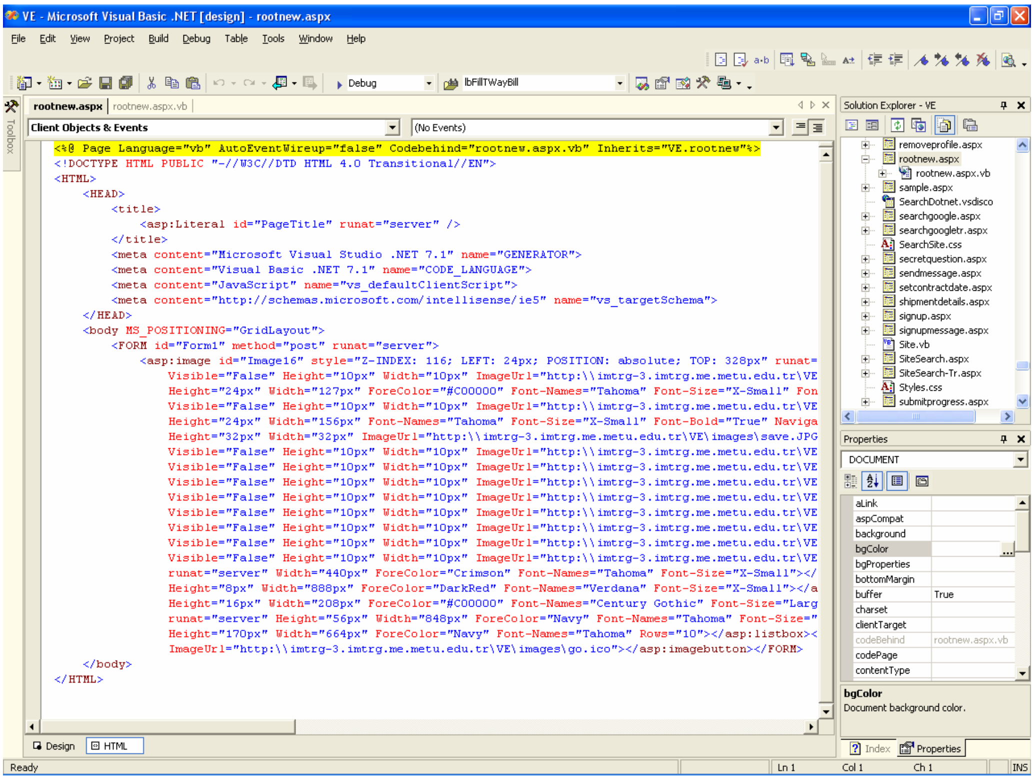


Figure 5.24 Typical contents of a web form

Web form is an ASP.NET technology that is used to create a programmable web page. It is built on the Common Language Runtime, thereby providing a managed execution environment, type safety, inheritance, and dynamic compilation.

It should be noted that the Internet web server should have capability of executing .aspx extension web pages. IIS 5.0 is the most commonly used tool of Microsoft for large scale internet and intranet applications.

In ASP days, all ASP applications had to be developed using in-page coding (also referred to as inline coding by some programmers) because that was the only way to develop an ASP page. Often, the intermixed HTML and scripting codes in a large page become cryptic and difficult to read and maintain. Fortunately, ASP.NET provides a way out of this problem. The programmer may develop the html code in a file with an .aspx extension and then necessary code can be written in a separate VB code file. This practice is known as Code-Behind. Programmer develops an .aspx file where the layout of the controls is defined in a page, and code is included in a separate VB or C# class file. This mechanism separates the page layout design activities from the code development activities.

5.2.3.2.1 Built-in ASP objects

As the scripts are run on the server, there are a number of useful objects that the developer can get use of:

- *Request*. Retrieves the values that the browser passes to the server during an HTTP request.
- *Response*. Controls what information is sent to the browser in the HTTP response message.
- *Session*. Used to exchange information between ASP pages.
- *Application*. Used to manage and store information about the Web application.
- *Server*. Provides access to resources that reside on server.

Some of these objects are frequently used and therefore rather important, the use of them will be demonstrated in the following sections.

5.2.3.2.2 Reading requests

The request object contains five collections that can be used to extract information from an HTTP request. These are the following:

- *ClientCertificate*. The values of the certification fields in the HTTP request.
- *Cookies*. The values of cookies sent in the HTTP request.
- *Form*. The values of form elements posted to the body of the HTTP request message by the form's POST method.
- *QueryString*. The values of variables in the HTTP query string, specifically the values following the question mark (?) in an HTTP request. Values sent to an HTML page using the GET method are retrieved using the QueryString

collection.

- *ServerVariables*. The values of predetermined Web server environment variables.

Using QueryString Collection

When a user submits a form with GET method or appends parameters to a URL request, QueryString collection should be used to collect the information. For example, the following HTTP request is done:

```
http://product.aspx?product_id=20004&status=new
```

The information can be extracting from product page by:

```
Request.QueryString("product_id")
Request.QueryString("status")
```

Using Form Collection

When a user submits a form with the POST method, it is possible to read the values of the controls by Form collection. As an example the user can submit a form with the following HTML:

```
<FORM ACTION="submit.aspx" METHOD=POST>
<INPUT type="text" id=userid name=userid>
</FROM>
```

The userid variable that is submitted by the page can be read from the submit.aspx file by the following code:

```
Request.Form ("userid")
```

5.2.3.2.3 Responding outputs

The response object enables to control the information sent to a user by the HTTP response message. The following are some methods of the response object:

- *Clear*. Clears any buffered response.
- *End*. Stops the processing of a Web page and returns whatever information has been processed so far.
- *Flush*. Sends buffered output immediately.
- *Redirect*. Sends a redirect message to the user, causing the response message to try connecting a different URL.
- *Write*. Writes a variable to the current HTTP output as a string.

The following lines of code demonstrate the use of write method within a loop:

```
For Each Item In Request.Form
        Response.Write("Item")
Next
```

Also the following code demonstrates the use of Redirect method to send the user to another URL:

```
If Request.ServerVariables ("HTTP_UA_PIXELS") = "640x480" Then
        Response.Redirect ("lo_res.aspx")
Else
        Response.Redirect ("hi_res.aspx")
End If
```

5.2.3.2.4 Saving state data

ASP.NET enables to maintain state in a web application. It is possible to save two types of state information.

- Application state. Information is available to all users of a Web application.
- Session state. Information is available only to a user of a specific session.

This example shows to create and access a variable that is available to a user all through his session:

Session(“user_role”) = ”customer”

5.2.3.2.5 Server side processing

An ASP.NET file has an *.aspx extension. Typically, it contains HTML elements, server-side codes and client-side codes. As shown in Figure 5.25, when a user requests an ASPX page, the server retrieves it from the disk and then sends it to the ASPX Engine for further processing. The ASPX Engine compiles the server side codes and generates the page class file. It then instantiates the class file and executes the instructions to develop the response object. During the execution stage, the system follows the programmatic instructions (in the server-side code) to process the data submitted by the user. Finally, the server transmits the response object to the client.

In short, the major steps in processing a request for an ASPX page are as follows:

1. The server receives a request for a desired ASPX page.
2. The server locates the page in the disk.
3. The server gives the page to the ASP.NET Engine.
4. The ASP.NET Engine compiles the page and generates the page class. If the class had already been loaded, it simply provides a thread to the running class instead of regenerating the class. During compilation, it may require other code classes, such as code-behind classes and component classes. These are assembled during this step.
5. The ASP.NET instantiates the class, performs necessary processing, and it generates the *Response* object.

6. The Web server then sends the *Response* object to the client.

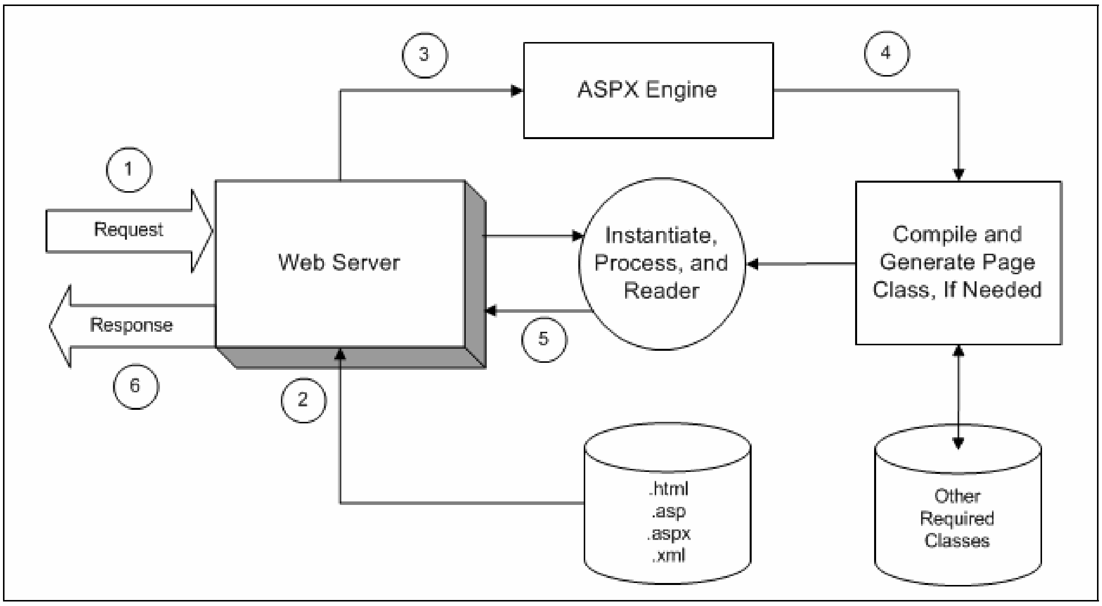


Figure 5.25 Major steps in serving an aspx Page

5.2.3.3 Testing and debugging

Testing is any activity whose aim is the measurement and evaluation of software attributes and capabilities. Hetzel, 1998 states that the purposes of testing are to:

- Gain confidence that systems may be used with acceptable risk.
- Provide information that prevents errors being made.
- Discover errors and system deficiencies.
- Discover what system capabilities are.
- Provide information on the quality of software products.

There are various levels of testing and different test methods that are appropriate for each level of testing (Rakitin, 1997). The various levels of testing form a hierarchy as illustrated in Figure 5.26.

At the bottom of the hierarchy is unit or module testing. The overall objective of this level of testing is to find bugs in logic, data, and algorithms in individual modules. After unit testing, next level is integration testing. Integration testing is intended to find bugs in interfaces between modules. The next is usually called validation testing. Validation is the process of evaluating software at the end of software development process to ensure compliance with software requirements. The objective of validation testing is to determine if the software meets all its requirements. As part of validation testing, regression testing is performed to determine if the software meets all its requirements in the light of changes and modifications to the software. Finally, there may be another level of testing, referred to as acceptance testing. Alpha and beta testing are variants of acceptance testing.

Similar to levels of testing, there are also several different testing methods. For example:

White box or glass box testing is a method of testing in which knowledge of the software's internal design is used to develop tests. It is used in unit testing and integration testing levels and performed by software engineers.

In functional or black box testing, no knowledge of software design is used, and tests are based strictly on requirements and functionality. It is used in validation, regression and acceptance testing levels and performed by software quality assurance team and software validation team.

Top-down, bottom-up, and outside-in testing are all examples of different methods for performing incremental integration testing whereby modules are integrated and tested based on their positions in the module hierarchy. It is used in integration level and performed by software engineers.

Act-like-a-customer (ALAC) testing is a method in which tests are developed based on knowledge of how customers us the software product. To maximize benefit to customers, defect detection and correction activities should focus on those bugs' customers are likely to find. It is used in validation, regression and acceptance testing levels and performed by software quality assurance team, project team and by the customer. The customers may also have their own tests.

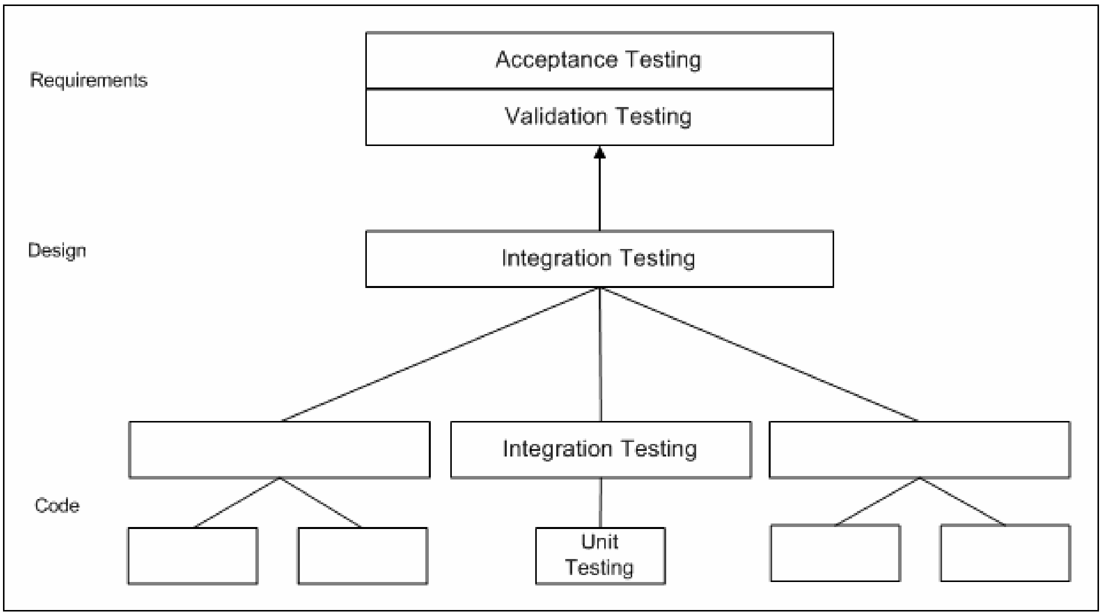


Figure 5.26 Testing hierarchy

Lastly, the most important point that one should keep in his mind when testing a program (Dijkstra, 1990):

> "Testing can show the presence of bugs but never their absence."

CHAPTER 6

VIRTUAL ENTERPRISE METHODOLOGY

The aim of this chapter is to address the third research question: *RQ3: How to set up & operate VEs including what to consider when preparing for it?* The aim of the VEM is guide (virtual) enterprise engineers in preparing, setting up and operating virtual enterprise. VEM provide guidelines for what to consider in terms of preparing for and establishing virtual enterprises as well as their breeding environment i.e. the enterprise networks (Sari et al, 2006a).

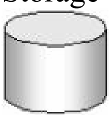
6.1 ICT Architecture for VE

Architecture is the design and contents of an organized environment for running a collection of objects. A wider definition would include within its scope the design of the total built environment, from the macro level to the micro level.

A modular and layered approach was used to define the ICT architecture for virtual enterprise systems with an n-tier client/server technology of Microsoft .NET framework in distributed manufacturing environment. This basis was drawn from the layered ICT reference architecture shown in Figure 5.1. ICT reference architecture is a resource containing a consistent set of architectural best ICT practices in distributed manufacturing environment.

It should be noted that the common VE platform is the one with which company specific platforms would interact and exchange released information with. VE participant's platform is the working environment and systems of the end-user through his/her organization specific applications and platforms. VE participant's platform in association with the VE platform is illustrated in Figure 6.1, and the contents of each module/layer described in Table 6.1.

Table 6.1 Description of layers in ICT architecture for VE system

Presentation 	In the foreseen implementation, this layer is served in part by several commercial products that present their own interfaces for the end-user to interact with. Alternatively, a simple web browser is used.
Application 	This layer is the main container for organization specific applications. In the implementation for VE, it can consist of different CAD tools, cost estimation tools, ERP modules etc. Furthermore, this layer is host to their different libraries and associated management applications.
Interoperability 	In essence, this layer enables the mapping of application specific syntactics and semantics to those of an agreed upon standard (e.g. IFC, STEP etc.). Some applications, e.g. ArchiCADTM possess a capability to do this on their own, while for others, this needs to be done through converters. This layer enables the exchange of product data between enterprise's layer and VE platform within STEP (Amaitik, 2005).
Communication 	This layer enables the communication between (the ICT systems of) a specific VE participant and the common VE platform. The Internet, data communication and exchange protocols, etc. form this layer. Within this research, the focus is on the Internet.
Access 	This layered provides control access to the VE platform. Different user specific profiles are furthermore provided enabling content viewing in different forms (when a query is made). In a similar fashion, information filtering is provided.
Service 	A simple document/file management service is provided. Distribution of product data to participants is done once the product has been announced.
Storage 	This layer forms the container where different types of data and information are stored. A simple file server is used for the storage of documents (e.g. CAD drawings, product and membership contracts, etc.), while the SQL Server is used for data storage.

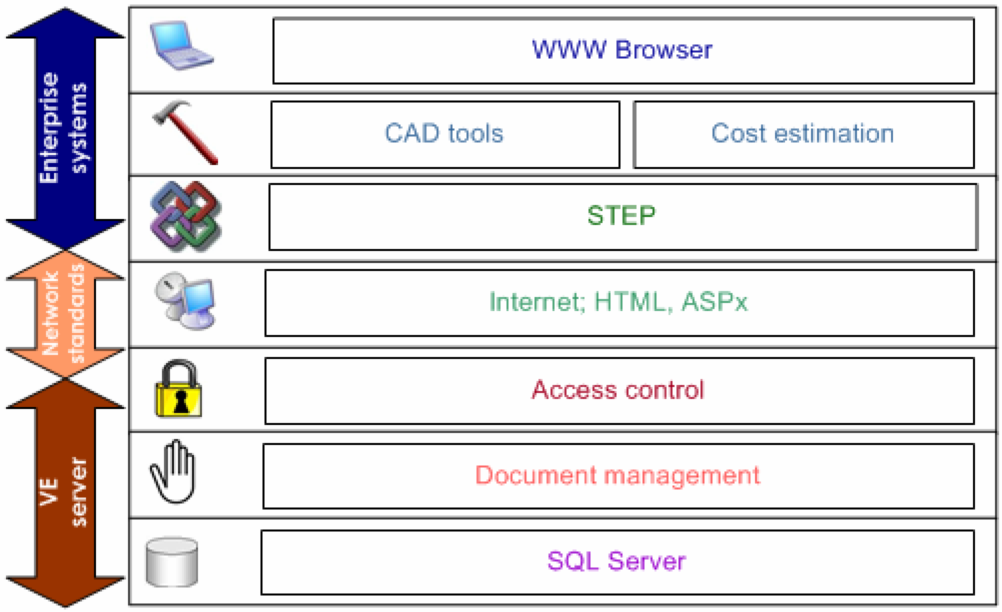


Figure 6.1 Layered ICT architecture for VE system

6.2 VE Methodology

6.2.1 Analysis phase

6.2.1.1 IDEF0 functional analysis

The most generalized IDEF0 representation of the VE system, called the top-level context diagram node A0, is given in Figure 6.2. It performs the function A0 of "Constitute Virtual Enterprise System". In this top-level node, the basic inputs are *product demand from the customer*, *business opportunity*, and *job tender from the interested enterprises*. Controls of the function are by *VBE membership* and *VE requirements*, and the mechanisms used are the *system actors (VE coordinator, VBE member, and customer)*. Through this basic function, *virtual industry clusters* will be formed, then the *product* will be established, and the *analyzed or processed data* will be sent to the system database.

The IDEF0 subsystem area A0 is then subdivided to define further each of these areas. Decomposition of Node A0, page 1 is given in Figure 6.3. The main purpose of *Set up VBE* function is to prepare and manage the life cycle of VEs. It establishes mutual agreements among its members on issues such as common standards, procedures, intellectual property rights, and ICT, so that these time-consuming preparations can be significantly shortened when a customer request arises, and a VE is put in place. The VBE or network should be seen as a potential from which different VEs can be established in order to satisfy diverse customer demands. The network will seek out and await customer demands, and when a specific customer demand is identified the business potential is established by forming a VE.

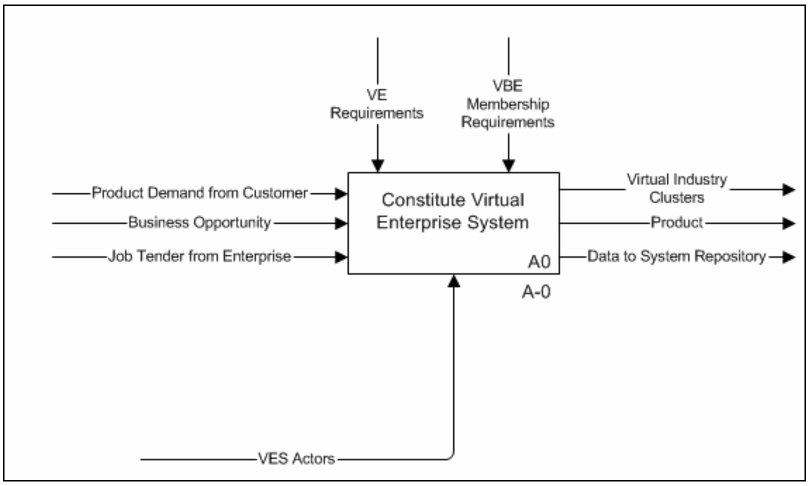


Figure 6.2 Top level context diagram, node A0

When a business opportunity is received, VBE will be settled and network members will be identified to respond quickly to rapidly changing markets driven by customer-based valuing of products. The decomposition of the *Set up VBE* (Node A1) will be explained in the preceding sections.

Please note that, in the set up of a VE from an enterprise network or VBE, the rather 'loose' relations that can exist among enterprises in a VBE become 'solid' in a VE. In a VBE, there is no notion of a customer-specific product or project. On the other hand, the VE is set up with a specific purpose in mind, i.e. a specific project delivering a specific product for a known customer. The configuration of the VE comprises the definition of the tasks/roles of individual enterprises and the relations among them. Concrete agreements are made regarding tasks, schedules, payments, and so on, which are detailed in contracts and project plans.

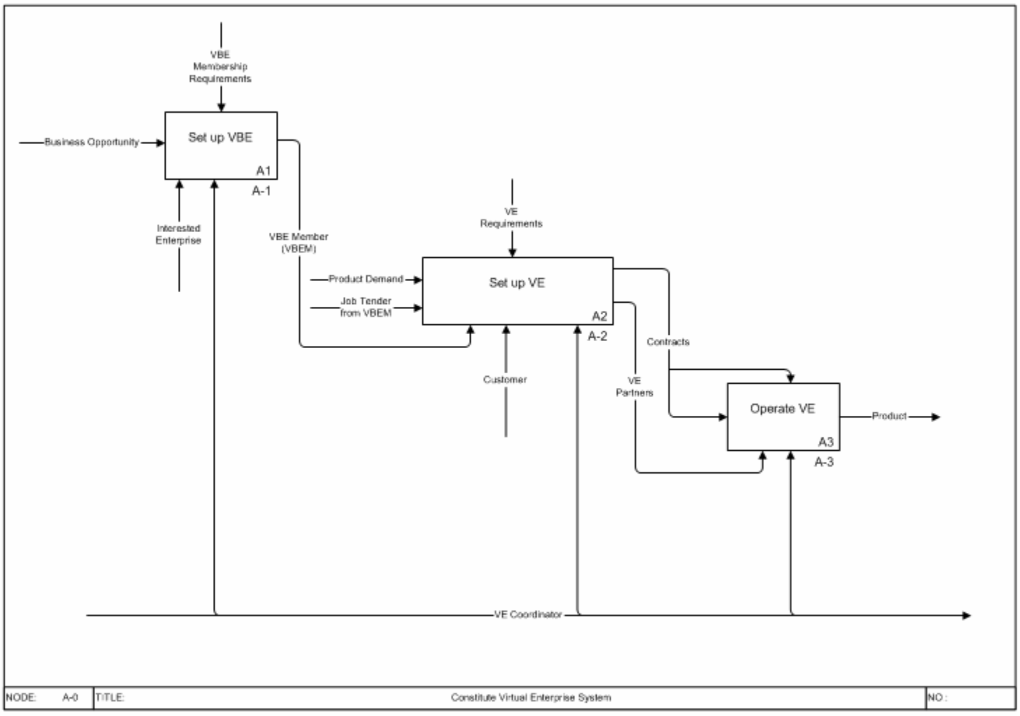


Figure 6.3 Decomposition of node A0 -page 1-

Also note that VBE members are the enterprises who are selected from interested enterprises willing to join the breeding environment. On the other hand, VE partners are derived from VBE members in order to form a customer-oriented VE. These partners are a subset of the VBE members.

VEs are based mainly on the integration of competencies among independent enterprises, providing a product or service that they are individually unable to offer on time, at lower cost than any of the cooperating partners would be able to achieve by themselves. The most important stage of the VE formation is to set up VE answering customer needs. At this stage, product demand from customer or job tender from enterprises fire the *Set up VE* function. The main outcome is being the formation of virtual enterprise.

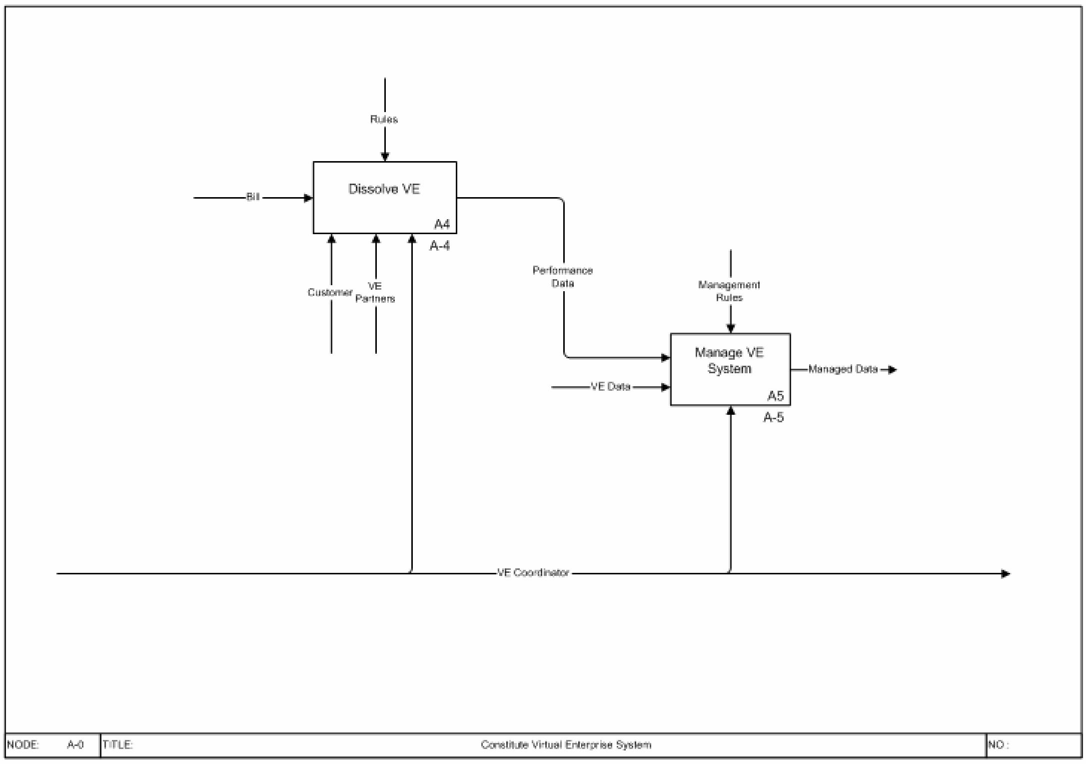


Figure 6.4 Decomposition of node A0 -page 2-

Once the VE has been set up and configured it goes into *Operate VE* and begins to fulfil the customer needs. Depending on the situation in question this includes design & engineering, construction and after sales service activities. The focus of system is, as mentioned, how to do inter-enterprise management of delivery processes. Thus monitoring of the VE with scheduling of progress submissions are sub-activities which will be elaborated in the further sections.

Decomposition of Node A0, page 2 is given in Figure 6.4. Once the product has been handled over to the customer and all paperwork and payment are taken care of it is time to *Dissolve VE*. However, before close it down the partners and VE members should give some time to capture their experiences learn in this respective project. These experiences will be stored in the documents repository for future use. Partners' performances during the operation of VE will be assessed for further partner selection processes. Node A5 represents the *Management of VE System*. Management of VE includes all types of management tasks and levels as known from traditional management of conventional enterprises. This includes, e.g. direct and indirect monitoring, and operational, tactical and strategic level decisions.

The decomposition of the *Set up VBE* process (Node A1) is given in Figure 6.5. The aim of the *Determine VBE Mission* phase is to identify the overall purpose of the breeding environment including the VBE type, the nature of need for investigating the VBE, and its boundaries in relation to the internal as well as the external environment. The main outcome of this phase is the clarification of the overall purpose of the environment including which type(s) of market(s) it shall address and with which type of products and consequently which types of competencies (i.e. partners) are required to fulfill the identified needs.

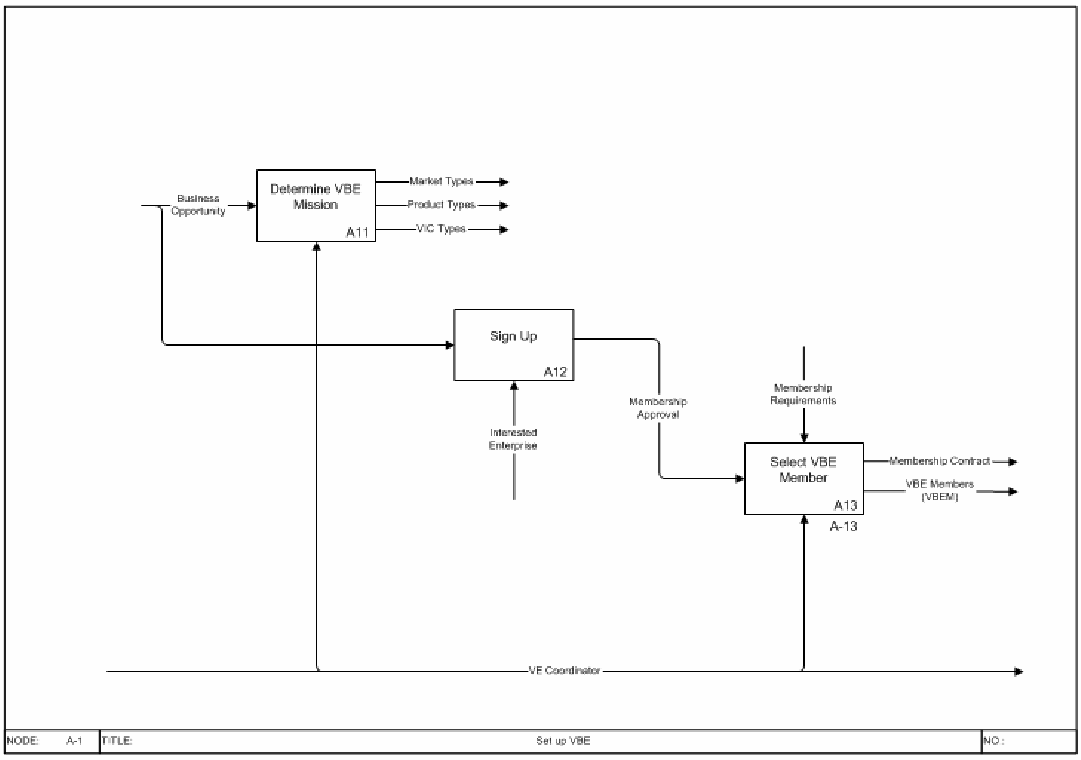


Figure 6.5 Decomposition of node A1

The next phase will be S*ign Up* of interested enterprises who are interested in to become a member of VBE. Once the need for the breeding environment has been identified & interested enterprise are approved for membership, the next stage is to

Select VBE Members having common goals with VBE and willing to participate in breeding environment. If an interested enterprise can satisfy VBE requirements, it will become a VBE member. The enterprises involved in a given breeding environment are normally registered in a database, where their core competencies and other attributes are declared.

The aggregation of VBE members from diverse industries, with well defined and focused competences, with the purpose of participating in virtual enterprises will form virtual industry clusters (VIC). As an example, two members dealing with machining will form machining cluster.

Furthermore, it can be relevant to look at the ICT infrastructure of an interested enterprise, e.g., is the enterprise using applications that you are able to interface and, likely important, is the partner willing to share the level and type of information you want access. A way to facilitate inclusion of VBE members is to develop VE tools applicable to member not having a high ICT maturity level, such as e.g. VE system, which as a minimum only requires a computer with Internet access.

Decomposition of Node A2, page 1 is given in Figure 6.6. The first phase; Node A21 will be S*ign Up* of candidate customers who are interested in to become a member of VE and give a product demand. Once the customer has been approved for membership, the next stage is Node A22, to *Select Customers* having common ICT infrastructure with VE. If an interested enterprise can satisfy VBE requirements, it will become a VBE member.

Next stage will be the Node A23; *Identification of Customer Requirements* in detail. Once a customer is registered to the system and demands a product, the system expects the customer to upload the product drawing in the required drawing format and to give detailed technical specifications concerning all features of the product. VE coordinator will examine the product demand and carry out a low level feasibility study. In this study, the feasibility of the product realization with the available resources will be determined. According to the result of this study, the coordinator will either accept the product demand and decompose the product into the relevant tasks or reject the product demand. If the product demand has been accepted, the coordinator should need to detail the customer requirements to get a comprehensive basis for the product data. Designers will have to re-establish the required technical drawings using their own CAD-tools.

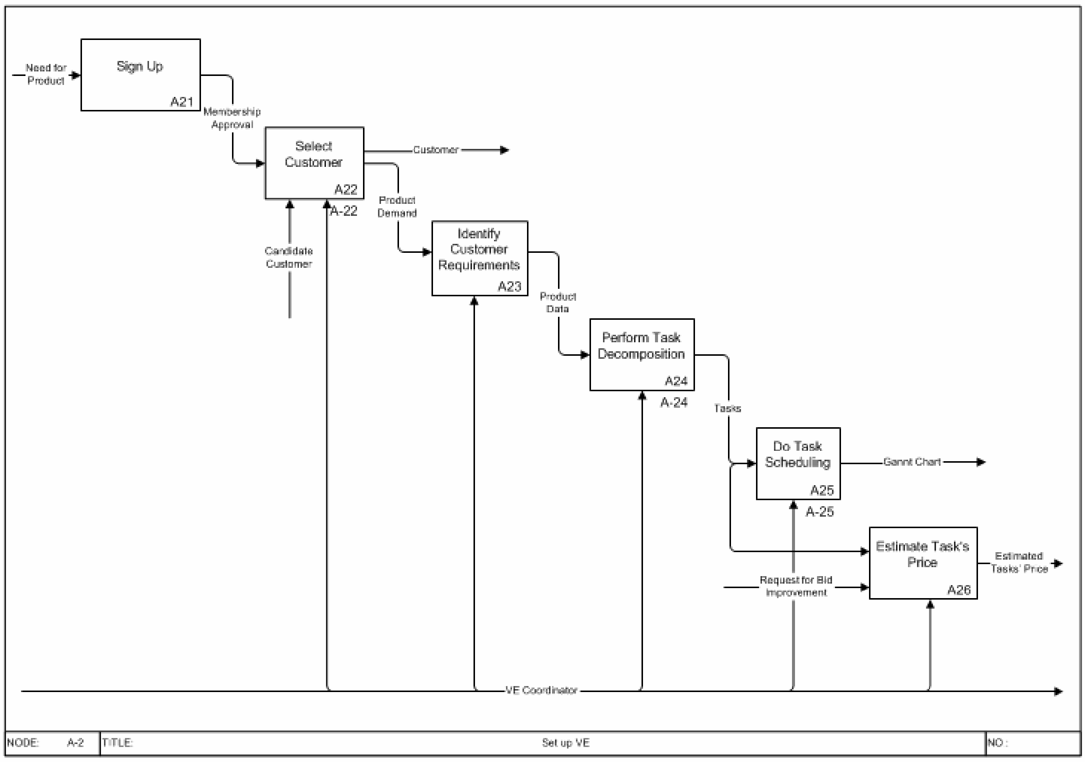


Figure 6.6 Decomposition of node A2 -Page 1-

Node A24, *Perform Task Decomposition* is capturing the decomposition of the product into deliverables, identification of relevant tasks and different types of dependencies between two tasks for accompanying partner selection, i.e. the partners that are responsible for those respective tasks. Successive decomposition is needed to determine a set of tasks in which every task can be assigned to single organization satisfying task requirements (this organization can usually be a partner firm or another VE seen as a single firm). A task assignable to a capable & responsible partner will no longer be decomposed. At this stage, task concept granularity is variable. For each task, a description will be given to the members about the task announcement with the skills required of the VE, task availability and the deadline. Whenever a new product is announced, system automatically sends a notification to the member for the announced task via e-mail.

In Node A25, *Do Task Scheduling*, a detailed scheduling of the tasks needs to be done based on customer requirements. As an essential element in setting up VE, task scheduling aims to ensure that the tasks are completed on time. It includes definition and sequencing of tasks. Tasks due dates and different types of dependencies between successive tasks such as finish-to-start, finish-to-finish, start-to-start, start-to-finish will be determined (Finish-to-start means that the previous tasks have to be completed before the next one can begin, e.g. a part have to be manufactured before it can become a sub-assembly of the next partner).

For each task, a *Cost Estimation* procedure will be carried out in Node A26. In this procedure, the coordinator should estimate an approximate price for the each task. Note that, this estimated price will play a crucial role in the bidding process. In other words, members will be forced to propose a task price lower than given estimated price. Thus, it will represent the upper limit of the task price.

In Figure 6.7, decomposition of Node A2, page 2 is given. In Node A27, VE coordinator should announce the tasks and *Negotiate with Members* if their skill, resources, availability and cost bids are able to respond to the customer request, and then the coordinator should collect, evaluate & rank bids coming from members for each task. In Node A28, *Negotiate with Customer*, tender package related with the product demand will be offered to the customer. The product scope and extent and the delivery schedule defined in the tender have no more than one alternative. The tender require also negotiations with the customer in which the delivery package, schedule or price may be modified according to the request coming from the customer.

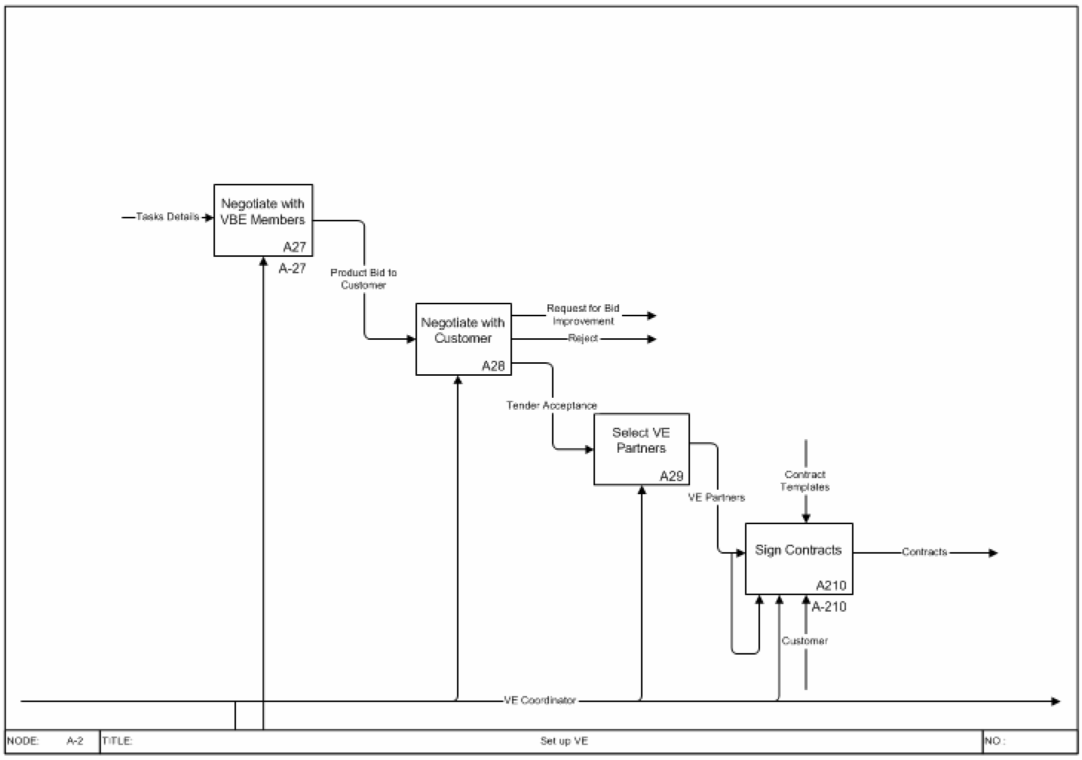


Figure 6.7 Decomposition of node A2 -Page 2-

If the customer accepts the tender package without requiring any improvement, then the coordinator will finalize the *selection process of partners*; Node A29.

A central element when preparing for and setting up an enterprise network is to *Select VE Partners* that will participate in the virtual enterprise. To participate in a VE, companies should basically possess two types of competencies, which are functional competence and alliance competence. Functional competence is the competence related to the product life cycle (PLC), therefore supporting the creation of the customer solution corresponding to the execution of the PLC. Functional competencies are described in terms of the type of solution and performance addressed by the competence, what capabilities it is composed of and how robust it is in terms of agility and sensitivity. Members' past performance rating being the indication of how efficient the partner is at performing a specific task will be evaluated as a functional competence in the selection process. Alliance competence is the competence related to the VE life cycle, representing the ability of a partner to enter into and participate in VEs. Two main elements of an alliance competence are the ability to manage and implement alliances and the ability to display alliance spirit and behavior. Thus task cost; how much the partner expects to be paid for the required task, caution cost which is the cost that the partner must pay to the VE if the partner decides the give up before the assigned task is finished and completion probability; the probability of completing the task by the given time period in the VE will be assessed as alliance competencies.

Completion probability is not easy to represent in quantitative terms. In order to create a quantitative value that can be used in the selection process, the probability is calculated using PERT. PERT uses three time estimate for each activity. Basically, this means that duration of each activity can range from an optimistic time to a pessimistic time, and a weighted average can be computed for each activity. Knowing the weighted average and variances for each task allows the VE coordinator to compute the probability of meeting pre-specified task duration.

Level of commitment is measured in terms of a caution cost which is the cost that the partner must pay to the VE if the partner decides the give up before the assigned task is finished and is secured in the form of letter of credit. Thus, the higher this value is, the more preferable for the VE. Note that once a VE is established, partners will be no longer competitors.

Overall performance of the partner company which is computed by taking the average of the partner performances shown during each previously executed task. Performance variability is based on delivery status of the task (late or early), quality of the completed task (% acceptable) and progress report submissions (late, early or none). A neural network model will be used to establish the performance score of each partner.

Analytical Hierarchy Process (AHP) will be applied to rank order the multiple partner proposals. AHP weights each of the user selection criteria and then weights the various proposal alternatives against each of these criteria. The result is a rank ordering of proposals that takes into account both the importance of each criterion to the user as well as the comparative evaluation of all alternatives against each of the selection criteria.

For each member bid, the attribute values (unit cost, caution cost, completion probability, past performance) retrieved from database are checked to see if they meet the constraints. If the values do not meet the constraints, then they are assigned the value zero. The overall score is calculated for all bids and the scores are ranked, where the highest score is at the top. VE Coordinator assigns the task to the VBE member who gets the highest rank.

Once the selection process of VE partners is completed, in Node A210 *contract will be signed* at this stage between the administrator-customer and partners-administrator after the product offer has been accepted by the customer. VE acts as a mediator between the customer and the partners. Both sides have a right to edit the contract template.

Once the Virtual Enterprise has been set up and configured it goes into operation and begins to fulfil the customer needs. Depending on the situation in question this includes design & engineering, construction and after sales service activities. Decomposition of Node A3, *Operate VE* is given in Figure 6.8.

Node A31, *Prepare Progress Submission Schedule* is together with Node A32, *Submit Progress* an important project management activity. During the operation of a VE, different actors (e.g. coordinator, customer) may want to know the status of the VE or part hereof and the progress made so far. Coordinator and customer may, for example be highly interested in whether the project works out as expected, in terms of physical work and cost. Therefore, coordinator should prepare a schedule for progress submission and the partners should submit their progress according to this pre-specified submission schedule.

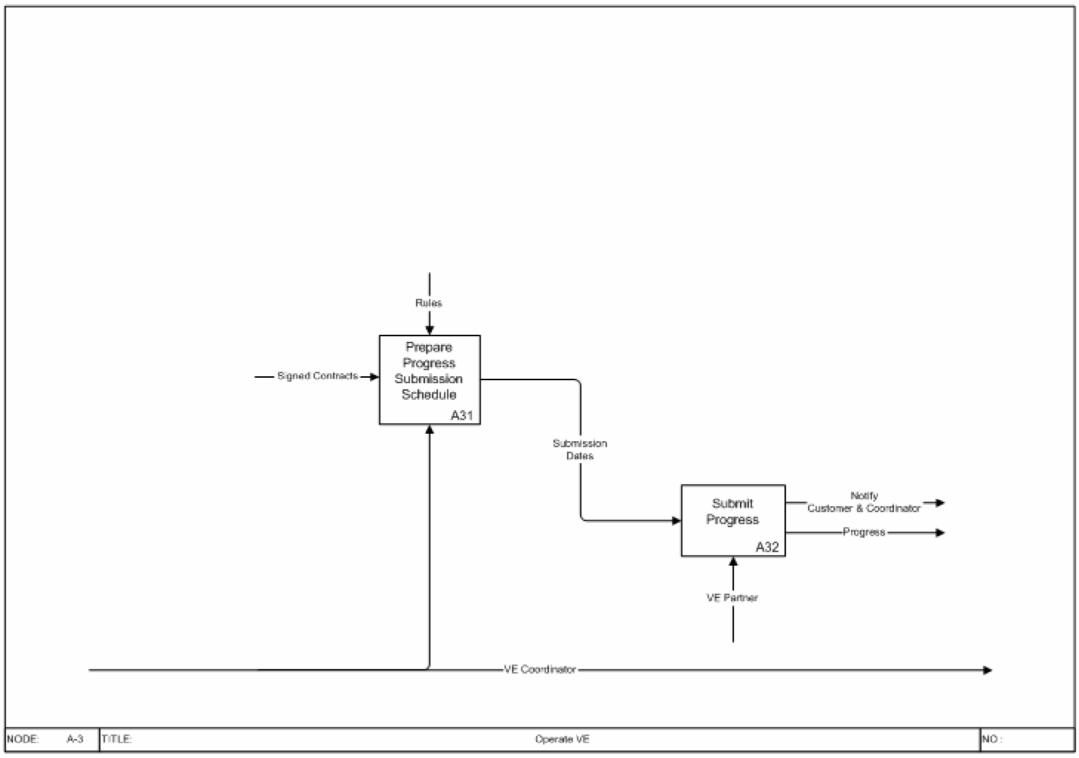


Figure 6.8 Decomposition of node A3

There are many things that can be monitored. However, the most important thing that should be monitored either by the system or manually by one of the actors is the progress at task level. General activities:

- Agree upon the type and level of progress reporting
- As a part of the VE set up, the coordinator shall determine which type of activities the progress should be reported. This includes the level of detail (which aggregation level), the reporting frequencies, e.g. every time a sub-task is completed and/or on a regular basis (e.g. once a week)
- Capture progress from partners
- During the operation of the VE, the responsible partners should report progress on their activities, e.g. reporting with a pre-defined time interval and/or every time a task is completed.

Once the product has been handed over to the customer and all paperwork done and payments are settled, it is time to execute Node A4, *Dissolve VE* as shown in Figure 6.9.

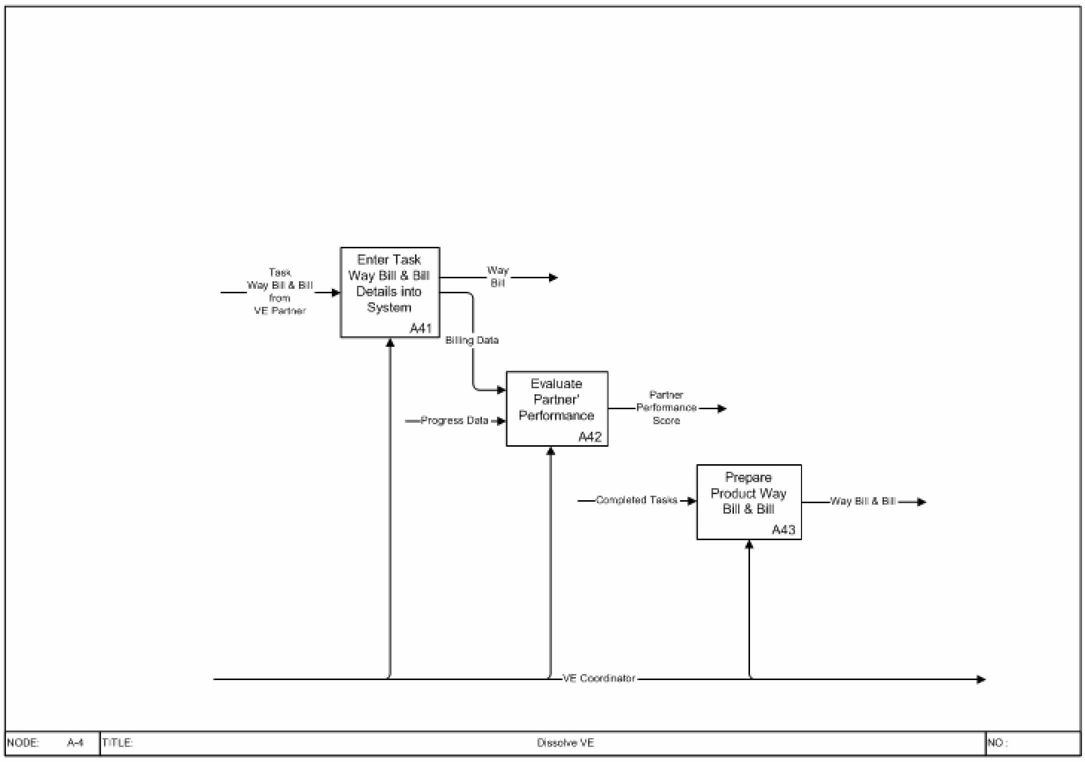


Figure 6.9 Decomposition of node A4

After the tasks have been completed, they will be handed over to the coordinator, the *way bill* and *bill* details will be entered to the system as stated in Node A41. In Node A42, *partners' performance* during the operation of VE will be assessed to be based for further partner selection processes. Before VE is closed down, the partners should give some time to capture their experiences learned in this respective project. These experiences will be stored in the documents repository for future use. Once all the collected information is stored in documents repository for further use, the project will be closed down and the partners go back to the virtual breeding environment and await new projects (VEs). After the product has been completed, it will be handed over to the customer, the final check and *billing* will be made as stated in Node A43, and the revenue will be divided according to the agreements between VE and its partners.

The decomposition of the Node A5, *Manage VES* is given in Figure 6.10 which includes *Manage VBE* and *Manage VE* processes, as stated respectively in Node A51 and A52. These processes include managing system mail accounts, messages, questionnaires, training & testing ANN, changing password, editing user profile, activating & suspending membership of users and other activities related with the VE management.

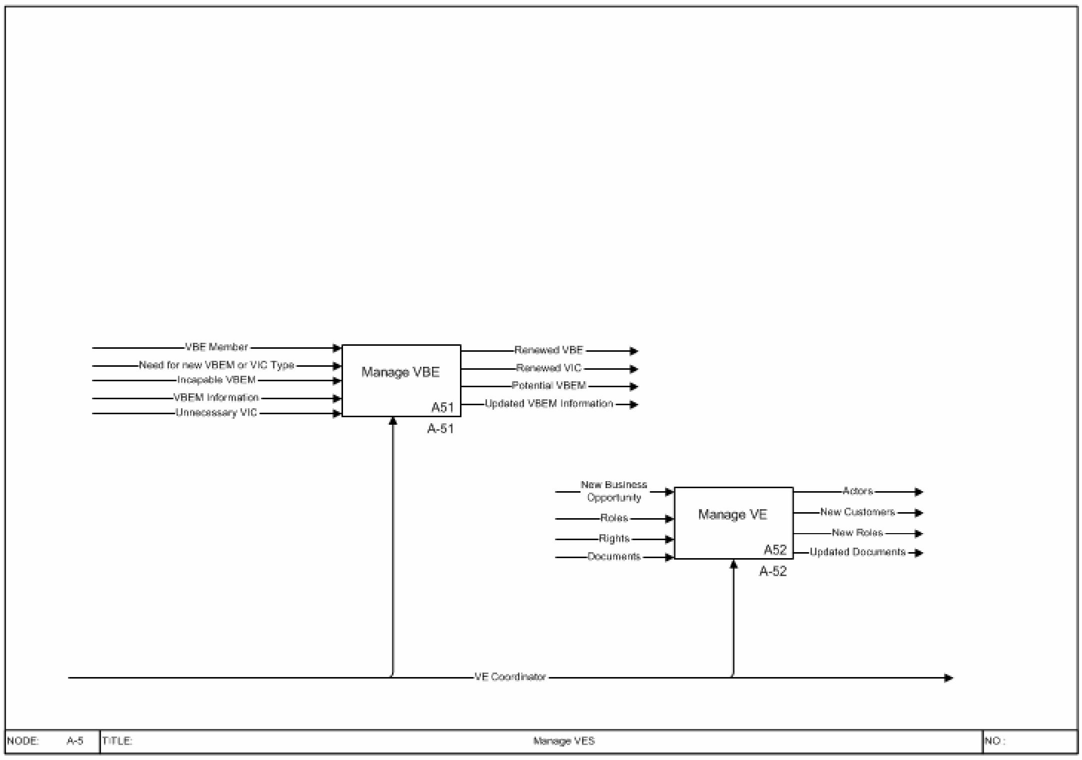


Figure 6.10 Decomposition of node A5

Understanding IDEF0 models of the VE is very important in the sense that substantial efforts are being focused on total system integration. It is the language of the virtual enterprise early in the design and implementation phase. Other IDEF0 diagrams can be found in Appendix A.1.

6.2.1.2 Use cases

Virtual enterprise system was divided into sections and for each section a use case diagram was drawn. Use cases defining the requirements of this section of the system and the actors interacting with them were stated. The interactions were modeled according to the authorization of the actors or users in the system.

Use case diagram of "Determine VBE Mission" module can be seen in Figure 6.11. The system is framed by a rectangle and name of the system is written top of the frame, which is "DetermineVBEMission". The use cases are: determine market types, determine product types, and define VIC types. The actor is: VE coordinator. The interactions of the actors with the use cases are represented with a line between them.

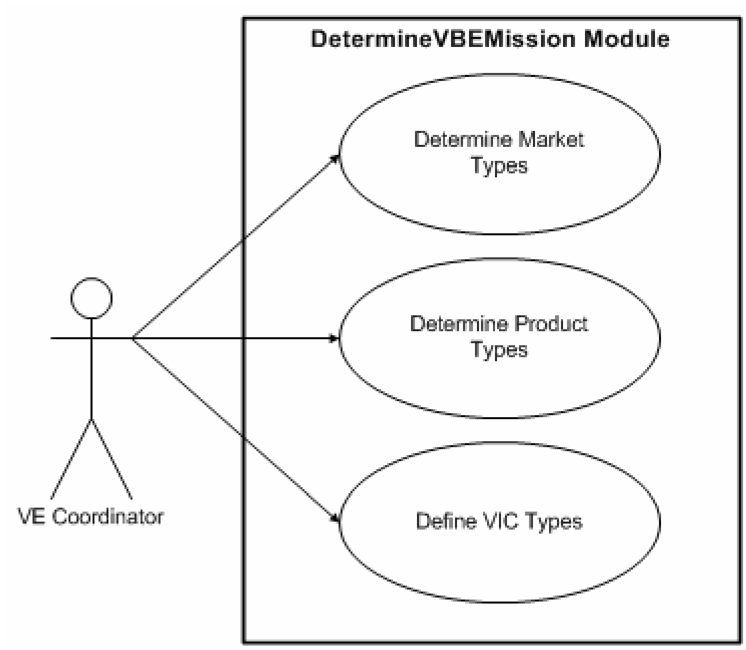


Figure 6.11 Use case diagram of "Determine VBE Mission" module

Another use case diagram related with setting up VBE and named as "SelectVBEM" is given in Figure 6.12. The use cases in this diagram are: examine enterprise details, visit company & interview, sign membership contract, and approve membership. The actors are: VE coordinator and interested enterprises. Note that, it can be seen from the use case that two actors can execute the same function synchronizely or asynchronizely. Please note that other use case diagrams can be found in Appendix A.2.

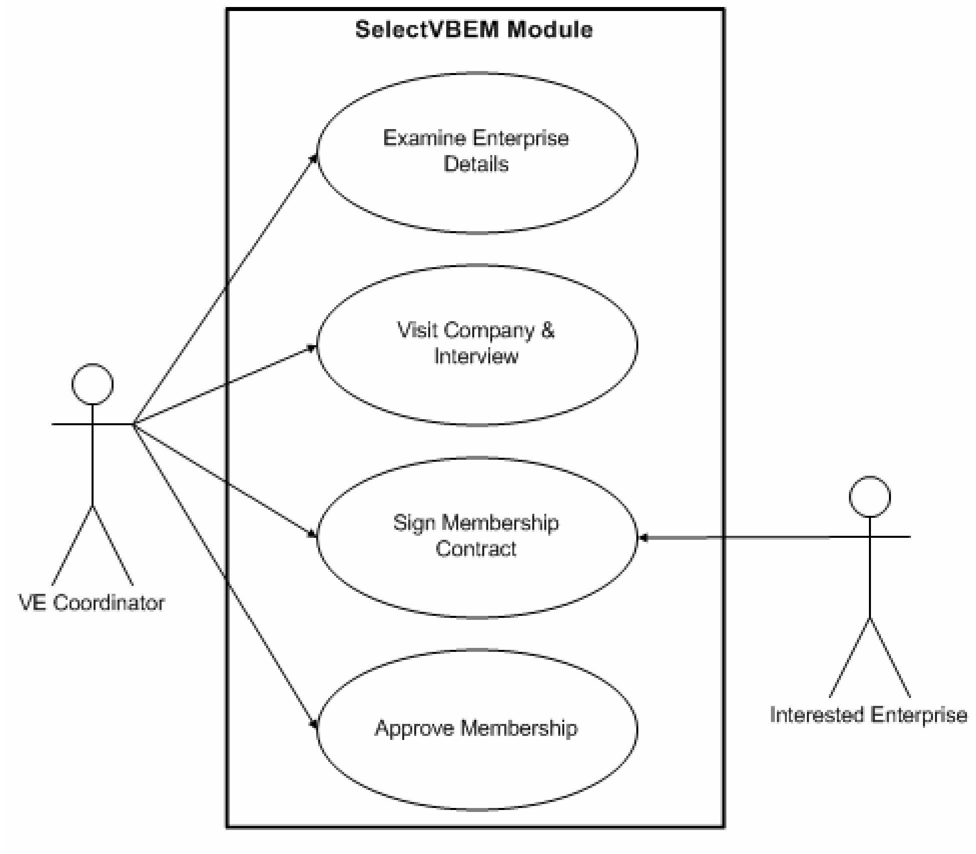


Figure 6.12 Use case diagram of "Select VBE Member" module

6.2.1.3 Activity diagrams

Once use cases are created, then the scenarios in the system were modeled using the activity diagrams. Flows of activities in some use cases were modeled to clarify these use cases. Activity diagram is very similar to the workflow diagram. Activity diagram of "Select VBE Member" use case can be seen in Figure 6.14. Selecting process starts with the retrieval of the interested enterprise decision on cooperation principles and data related with the enterprise's infrastructure from the database. Coordinator will eliminate the interested enterprise(s) who are not agreed with the cooperation principles and whose infrastructure does not satisfy VBE requirements.

If the cooperation principles have been accepted and infrastructure is appropriate, then the coordinator will visit the interested enterprise & verify the registered data. Interested enterprise will be eliminated in a case of failure of the verification. If the data being submitted is verified, then a membership contract will be signed between the coordinator & authorized person and then the coordinator will approve the membership approval. Thus the interested enterprise becomes a VBE member and its data will be updated. Other activity diagrams can be found in Appendix A.3.

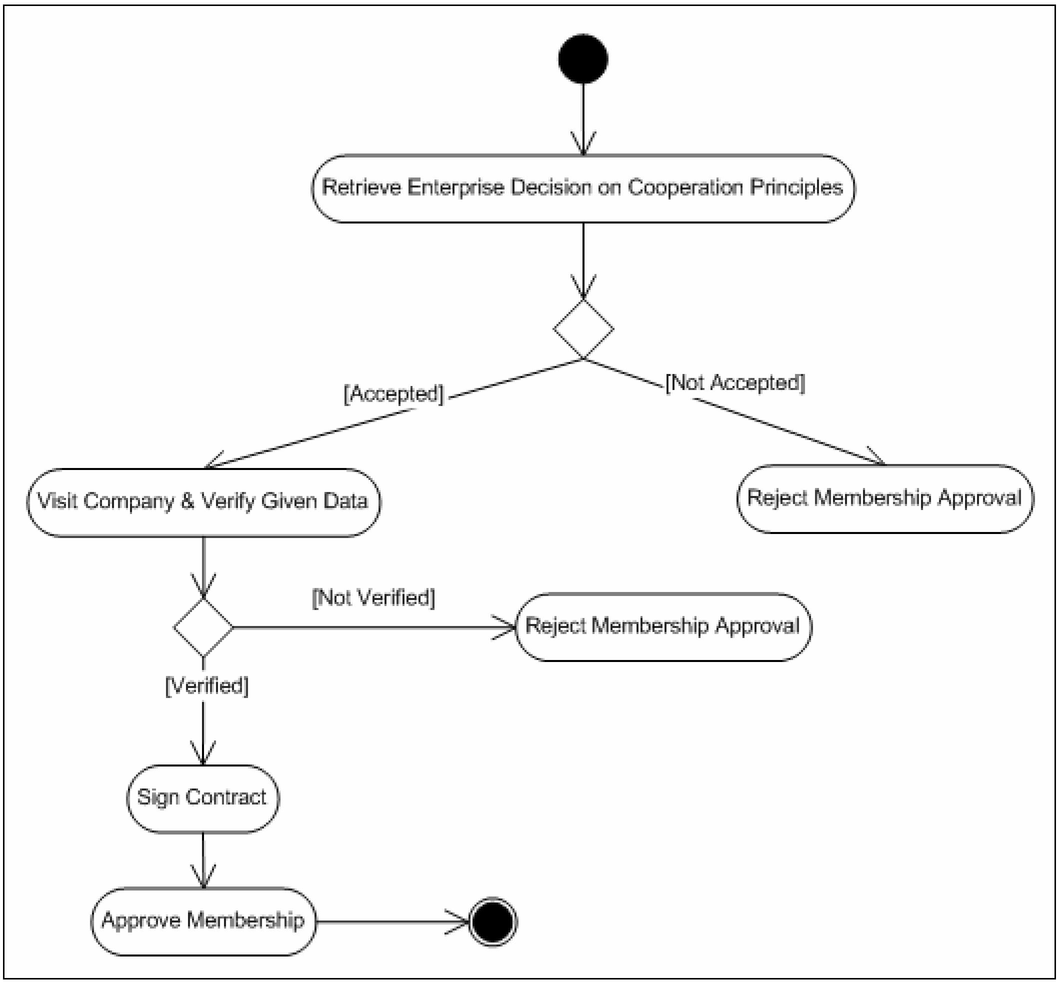


Figure 6.13 Activity diagram of "Select VBE Member" use case

Figure 6.13 illustrates the activity diagram of a VBE set up and VE set up, operation & dissolution.

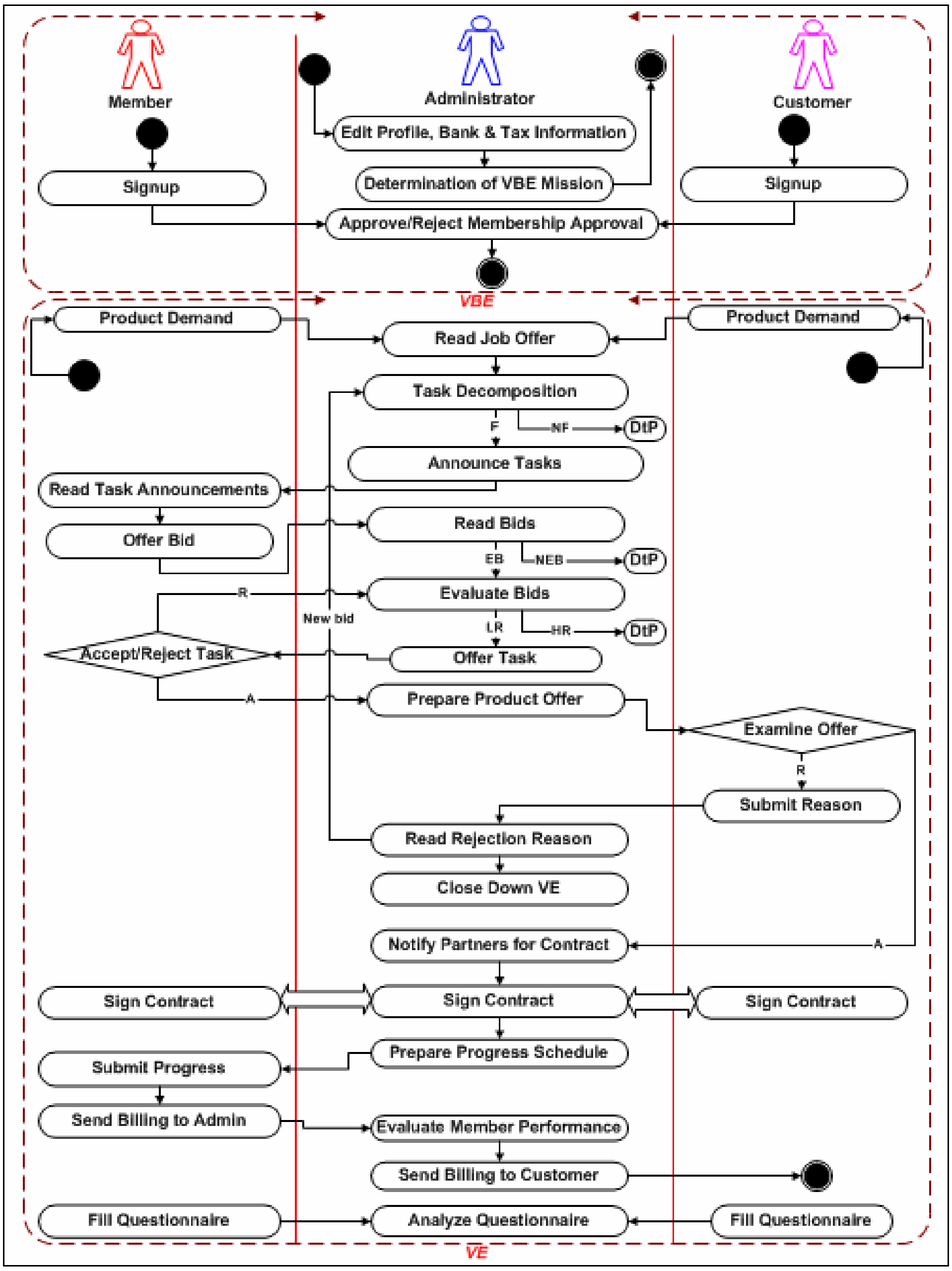


Figure 6.14 Activity diagram for VBE & VE

In addition to outlining the high level activities and key decision points the model indicates the entity types executing the activities/decisions. The figure distinguishes three actors: customer, member, and administrator or coordinator.

- The left part illustrates activities and decision points for the individual member enterprises;
- The middle part outlines coordinator activities and decision points;

- The right part addresses activities and decision for the customer.
- The upper part is related to setting up a VBE

The lower part addresses high level activities and decision points during set up, operation & dissolution of VE.

6.2.2 Design phase

6.2.2.1 Data-tier

6.2.2.1.1 IDEF1/X

One of the objectives of the virtual enterprise system design is localize the data storage thus by distributing the information increase the robustness of the system. However it is inevitable to store global data in distributed systems. Microsoft .NET framework provides access any kind of data resource, anywhere, thus to distribute or localize data in physical or logical meaning, is the basic feature of it. The relational data model for global information structure of the system is presented using IDEF1/X notation in Figure 6.15 (IDEF1/X, 1993).

It may be observed that the relational structure of the database also represents the inheritance and hierarchy between entities. From the figure one can inform that one *Market_Type* defined with a *Market_Type_ID* may have one or many *Product_Type*. A *Product_Type* consists of one or many *VIC_Type*, which is associated with one or more *Interested_Enterprise*. Each entry in *Interested_Enterprise* represents contact person & enterprise details. Once the membership contract is signed between the coordinator and the interested enterprise, *Interested_Enterprise* becomes to a member of VBE. *Product* may have one or many *Part* which may decompose into *Task* where each task will be assigned to a selected member. Task may have none or many *Task_Predecessor*.

Quality, delivery and progress scores which were selected as determinant factors effecting the performance assessment will be stored respectively in *Task_Quality*, *Task_Delivery* and *Task_Progress*. Performance score being the indication of how efficient the partner is at performing a specific task will be kept in *VBEM_Performance*. A neural network model is used to assess the performance of the partner companies taking into consideration the disciplinary and quality aspects. Training parameters (such as learning rate, momentum …), average square error resulted at the end of training process, network weights for every layer, and training data will be saved respectively in *ANN_Parameters*, *ANN_Error*, *ANN_Weights*, and *Cube_Performance*. Company details & contact person information related with the coordinator will be stored in *Admin_Details*. Tax, bank account & e-mail details of the coordinator will be kept respectively in *Tax*, *Bank_Account*, and *Server_Mail*. Name of the districts & cities available in Turkey will be kept respectively in *District* & *City* tables, whereas the countries available in the world will be stored in *Country* table.

Members offer bid(s) for the related tasks to take the advantage of being a partner in the virtual enterprise. Bids which are coming from the VBE members will be stored in the *Bids_From_VBEM* table. Overall scores being calculated for all bids via AHP will be kept in *Member_Scores*. *Product_ObjWeights* will keep the priority weights of alternatives with respect to attributes. *Customer* table is used to store the customer related information. *Contract_with_Customer* and *Contract_with_Partners* tables are used to store the data related with the contracts. Way bill & bill details related with the product will be saved in *Customer_WayBill* and *Customer_Bill*, whereas the data related with the tasks' way bill & bill will be kept in *Member_WayBill* and *Member_Bill* tables. In case of rejection of the tender package related with the product demand, the rejection reason and other details will be stored in *Customer_Acknowledgement* table. VE partners are required to report *Progress_Report* on their activities with a pre-defined time interval. Coordinator will schedule the reporting frequencies of the *Progress* and which type of activities the progress should be reported for. Incoming messages will be kept in *MsgInbox* whereas outgoing messages will be stored in *MsgOutbox*. The members will give some time to assess the system performance via questionnaire. Answers will be stored in the *Questionnaire* table for further analyze & improvement.

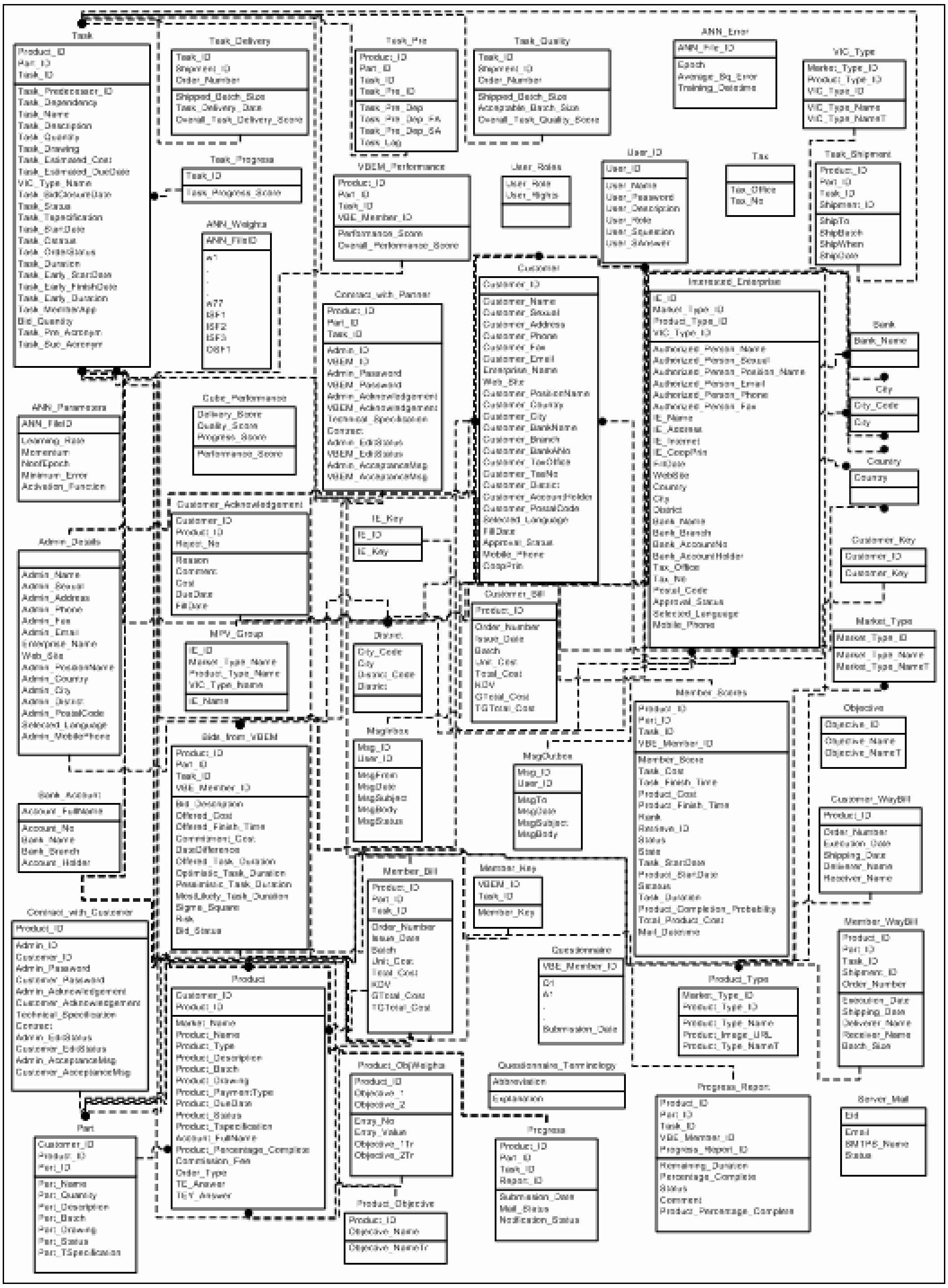


Figure 6.15 The relational data model in IDEF1/X notation

6.2.2.2 Presentation-tier

6.2.2.2.1 Authentication & authorization

In the system, all clients can access to web pages (aspx) with anonymous account that IIS placed in XP users which is namely IUSR_IMTRG-3. Programmatic authorization is used by programming by code for each page in the interface. The users are prompted to indicate a valid username and password to enter the web-site. By that, the identity of the current person is detected and a stored, also unattended entry to the system has been prohibited. Probably, the role of a user (enterprise) in a virtual enterprise is a major contingency factor for the business model, business processes, and applications it employs. A set of roles related to respectively the network and the VE were defined as:

- **Coordinator**: Coordinator is the leader of the network & VE, responsible for managing the relationship between the customer & the network and ensuring the fulfillment of the customer demands. Coordinator is also responsible to manage and maintain the portfolio of core competencies available in the network, to gather information about these competencies, and to configure VEs through selection of specific competencies required to fulfill the customer demands.
- **VBE Member**: VBE Members are the actors in the VBE & VE, i.e. the enterprises that supply competencies to the network.
- **Customer**: Customers are the actors having ability to give product demands.

6.2.2.2.2 Site diagram

For the presentation tier of the VE system, one major technique is used. It is the WWW based interface, which is developed as the system's site to communicate with the system on internet. The structure of the of the designed site diagram for the WWW based internet site is given in Figure 6.16.

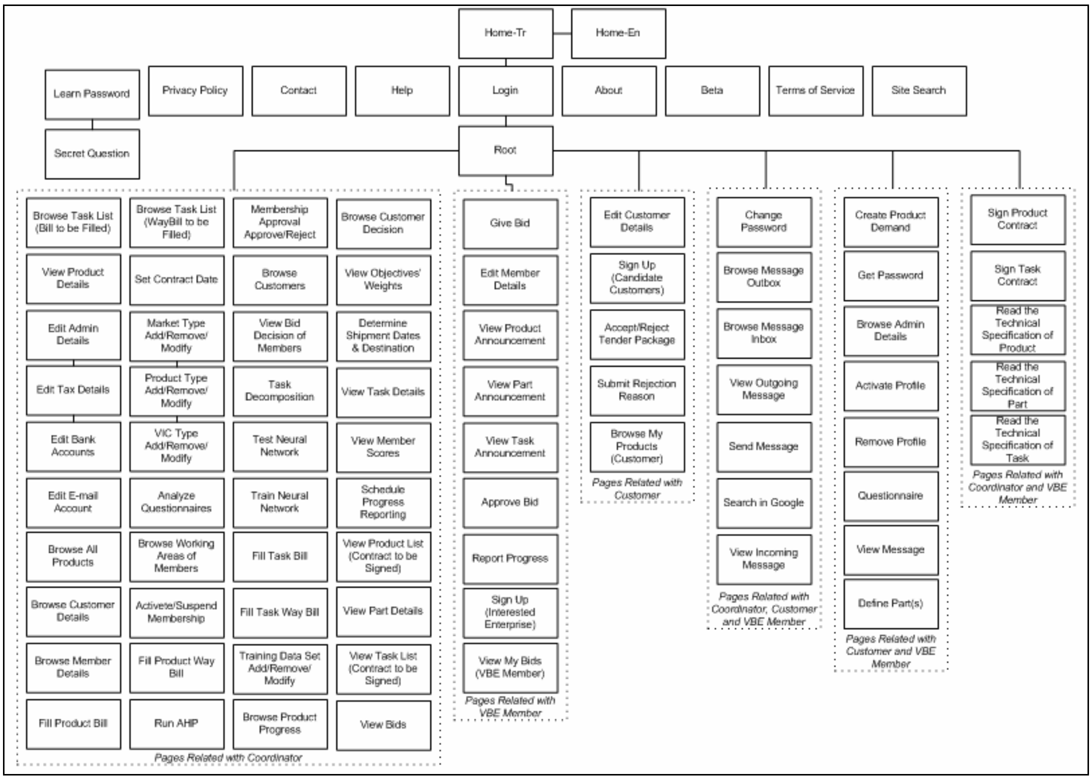


Figure 6.16 Site diagram

6.2.2.2.3 Class diagram

Using the textual description of the use cases, a noun-phrase analysis was conducted. The main objective is to identify the main actors (nouns) and processes (phrases) that are required to deliver the functionality of the system. Removal of redundancy, grouping of similar items etc. lead to simplistic identification of the main system modules (classes). The classes in the system are: Admin, AdminSendMessage, FillProductWayBill, FillBillofMember, FillProductBill, ContractwithCustomer, InterestedEnterprises, ManageCustomer, TestNeuralNetwork, MailAccount, FillWayBillofMember, TrainNeuralNetwork, InterestedCustomers, AnalyzeSurvey, Tax, ManageMember, BankAccount, InputData, SetContractDate, TrainNeuralNetwork, SelectPartners, ShipmentDetails, AnalyzeSurvey, ContractwithPartner, ProgressScheduling, ReadProduct, MsgInbox, Login, Logoff, ChangePassword, ForgotPassword, SecretQuestion, ReadPartD, ReadTaskD, MarketType, ProductType, VICType, MemberBidDecision, VBEMember, MemberProduct, SubmitProgress, ApproveBid, EditProfile, GiveBid, ActivateProfile, RemoveProfile, Product, GetIEPassword, SendMessage, UsabilitySurvey, Part, Customer, CustomerProduct, CustomerRejectBid, CEditProfile, MsgOutbox, GetCPassword, CustomerApproval, Task. Class diagram of the VE system is shown in Figure 6.17.

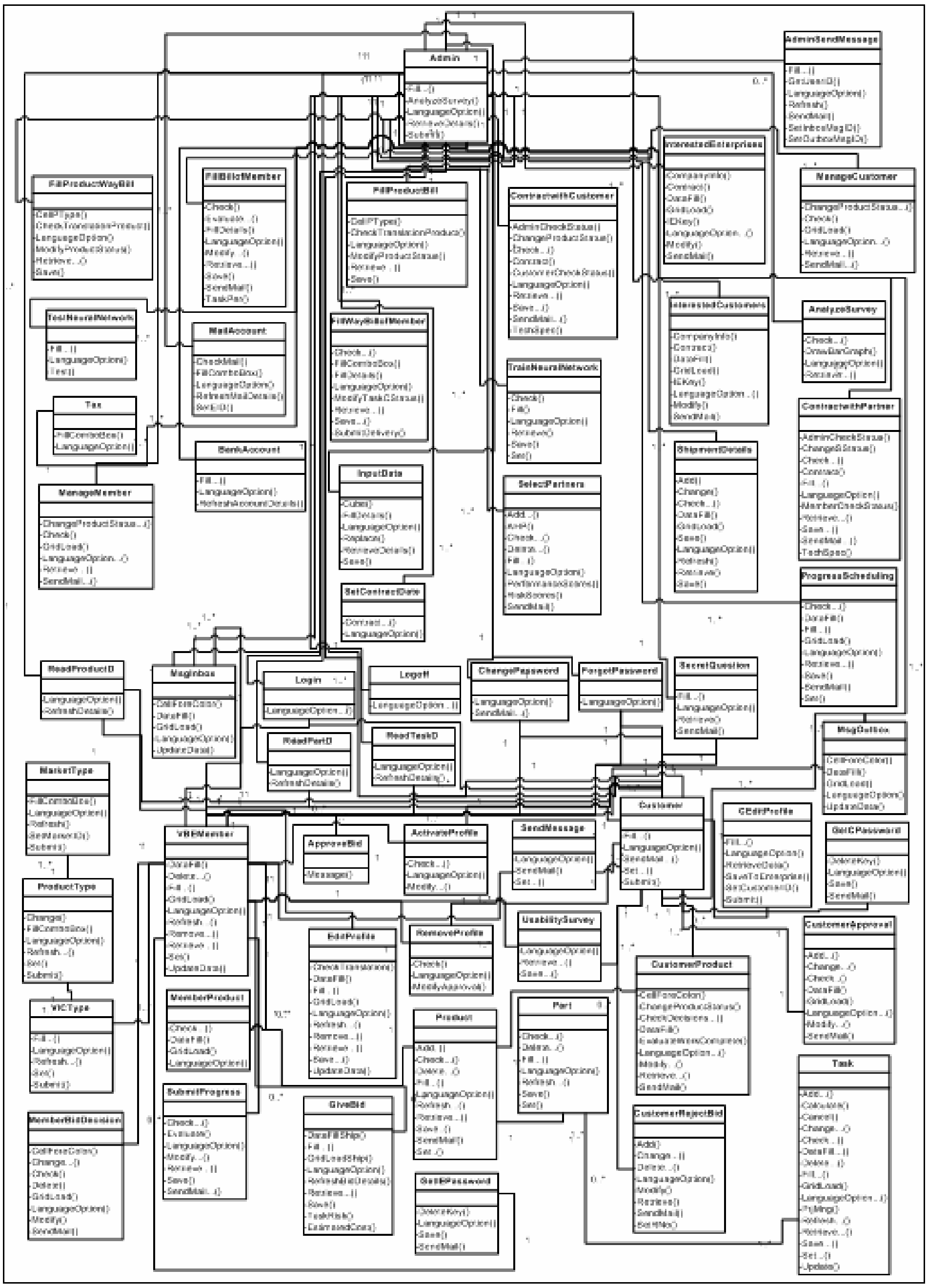


Figure 6.17 Class diagram of VE system

The duty of all of the classes is to communicate with the database. Methods of the classes are defined according to this duty. The "+" symbol preceding the names of the methods indicates that these methods are defined as public methods, which means that they can be reached by other classes. On the other hand, the "– " symbol means that the methods of the class can be seen in nested methods within the same class.

6.2.2.2.4 Sequence diagram

Using the identified system actors and processes, the use cases should be updated and sequence diagrams (at least one per use case) have to be generated. These diagrams typically describe the different interactions between the classes and the messages they exchange with each other. It is not a must to define all of the classes and complete the class diagram before going through the sequence diagrams. Developer does not have to draw the sequence diagrams for all the use cases. They are drawn to clarify the complex cases only. The sequence diagram of "Determine VBE Mission" use case can be seen in Figure 6.18. Other sequence diagrams can be found in Appendix A.4.

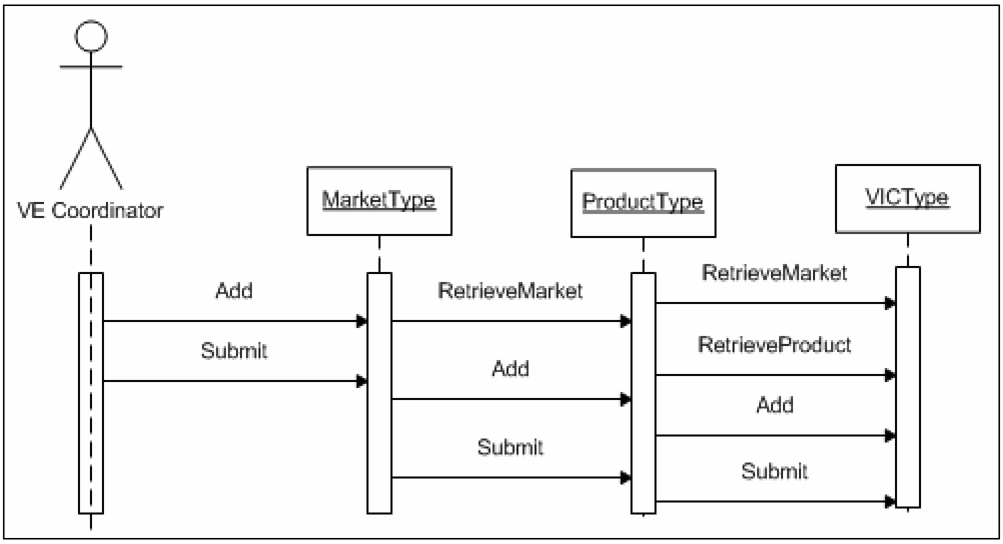


Figure 6.18 Sequence diagram of "Determine VBE Mission" use case

6.2.3 Implementation phase

Several implemented view from various parts of the system are given in from Figure 6.19 to 6.34.

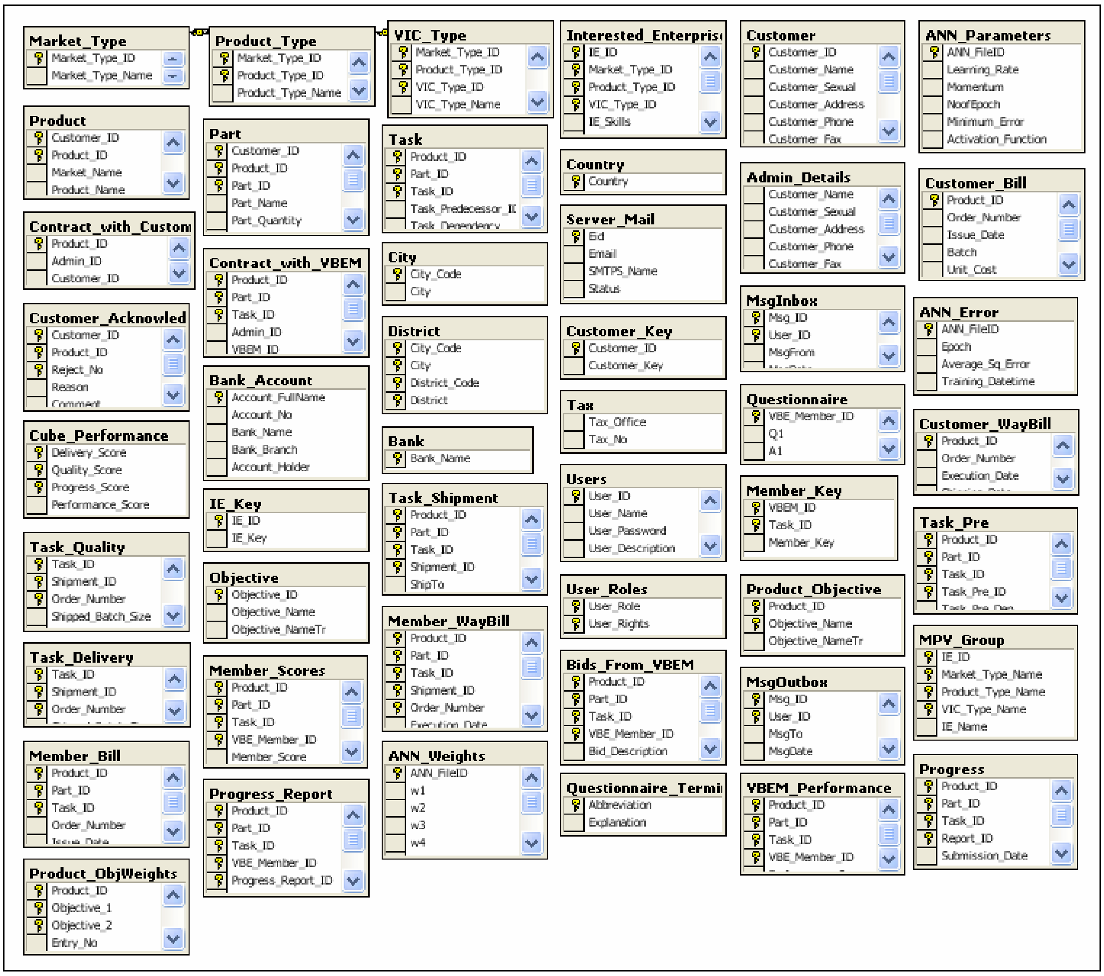


Figure 6.19 Design view of the VE database

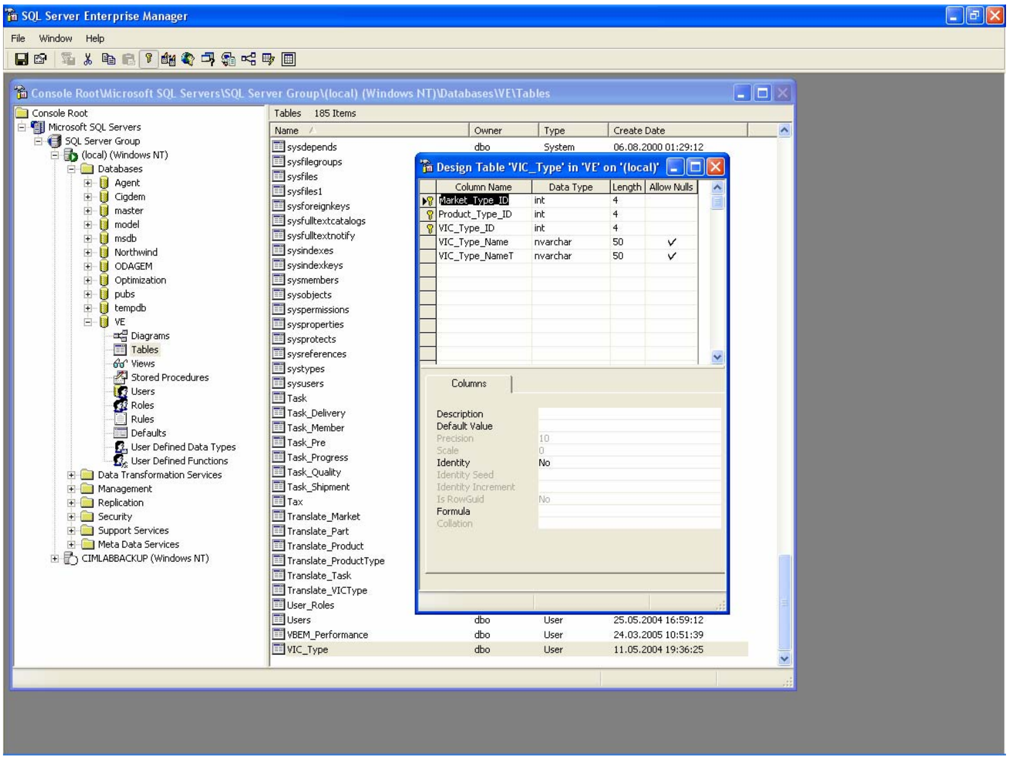


Figure 6.20 Design view of the table “VIC_Type” in VE database

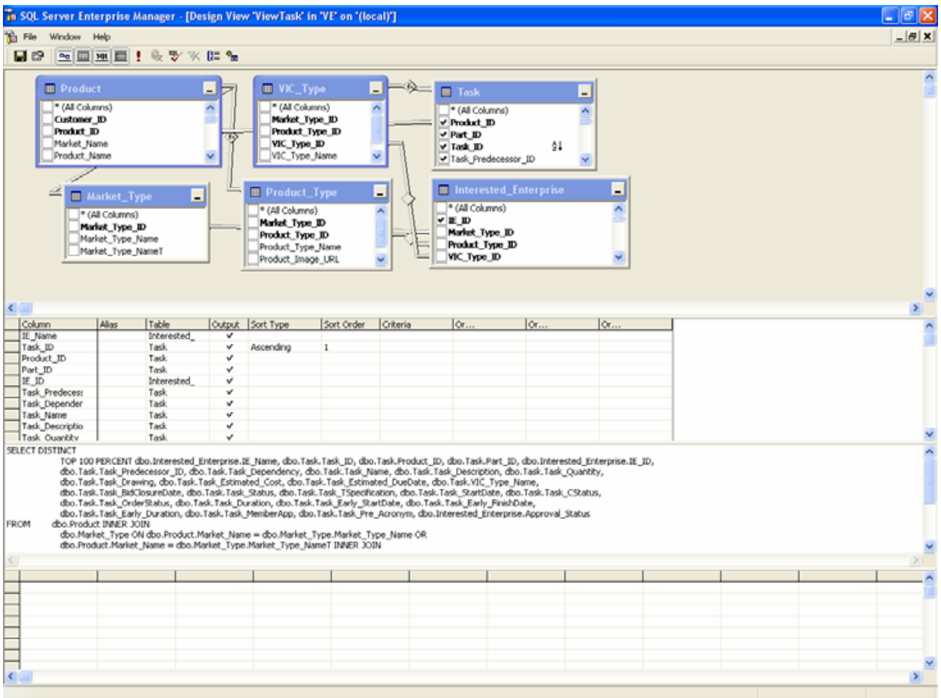


Figure 6.21 Design view of the view “ViewTask” in VE database

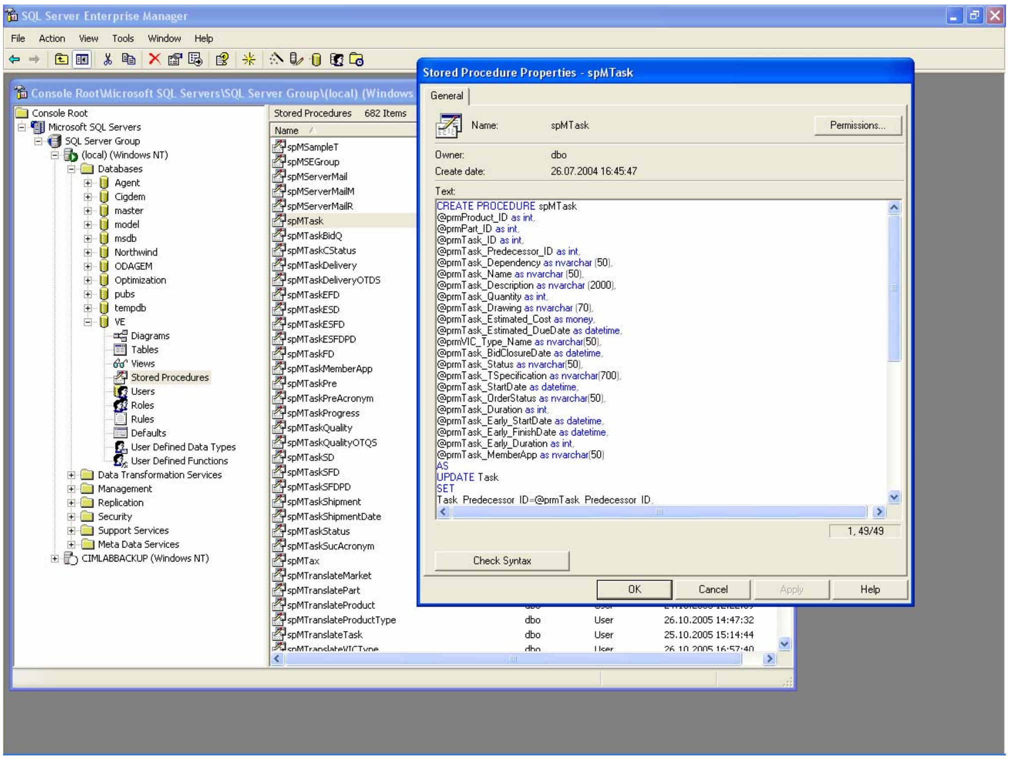


Figure 6.22 Design view of the stored procedure “spRTask” in VE database

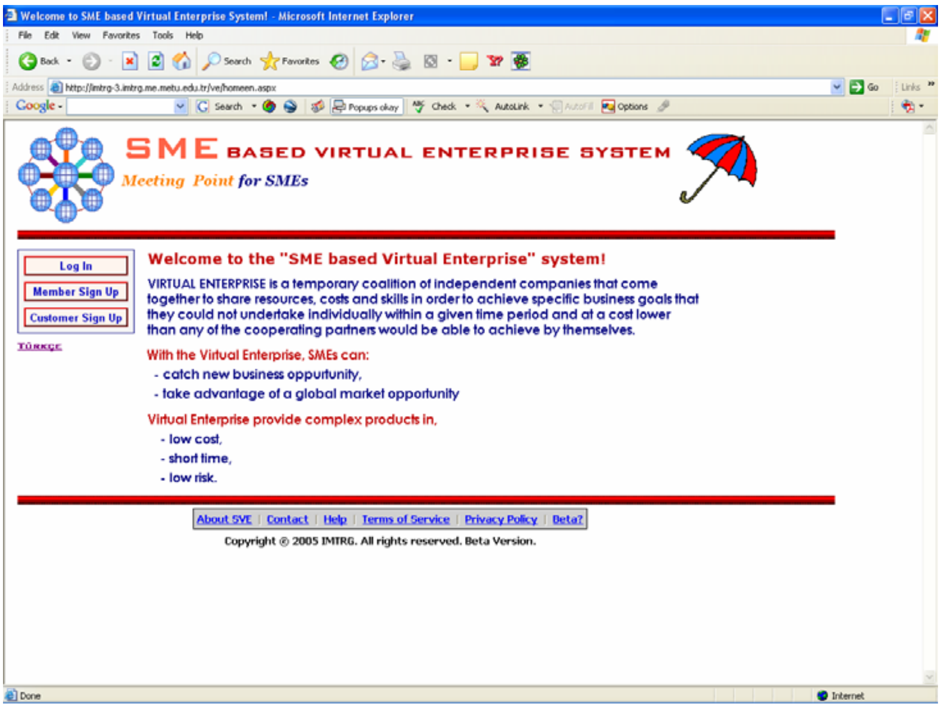


Figure 6.23 Home page

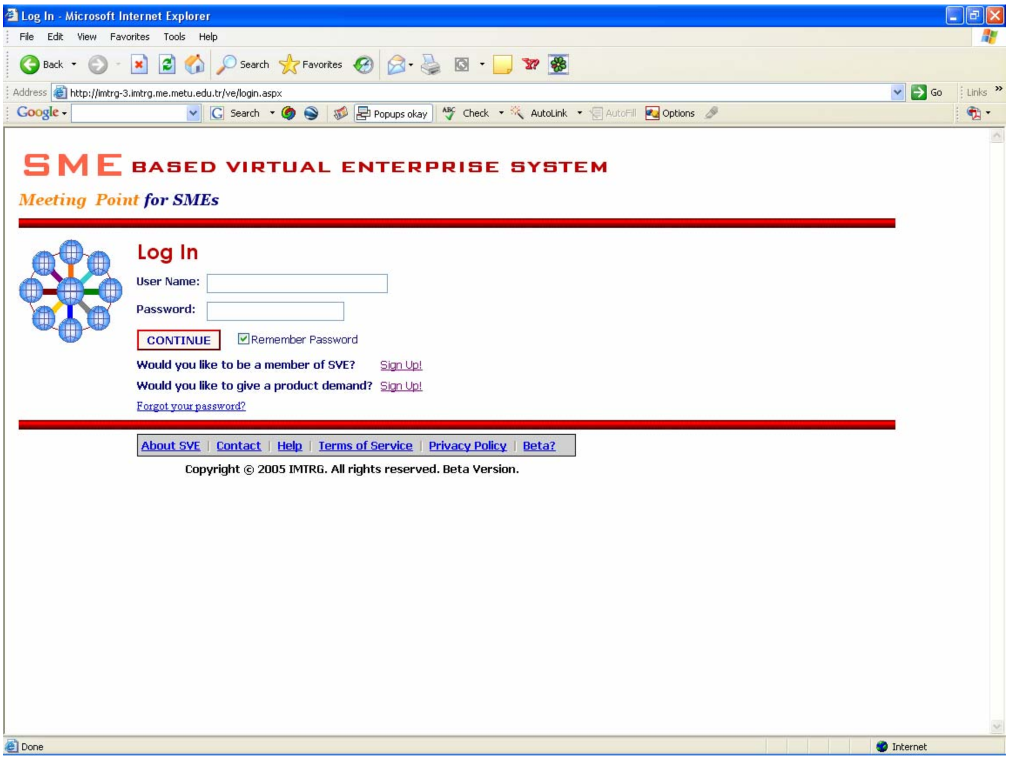


Figure 6.24 Log in page

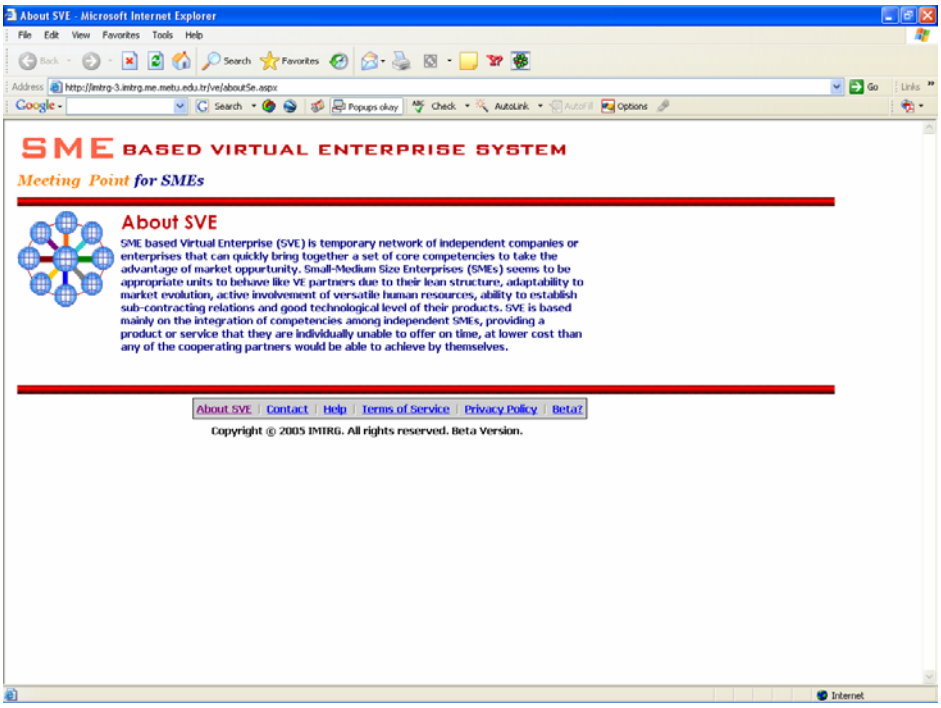


Figure 6.25 About SVE page

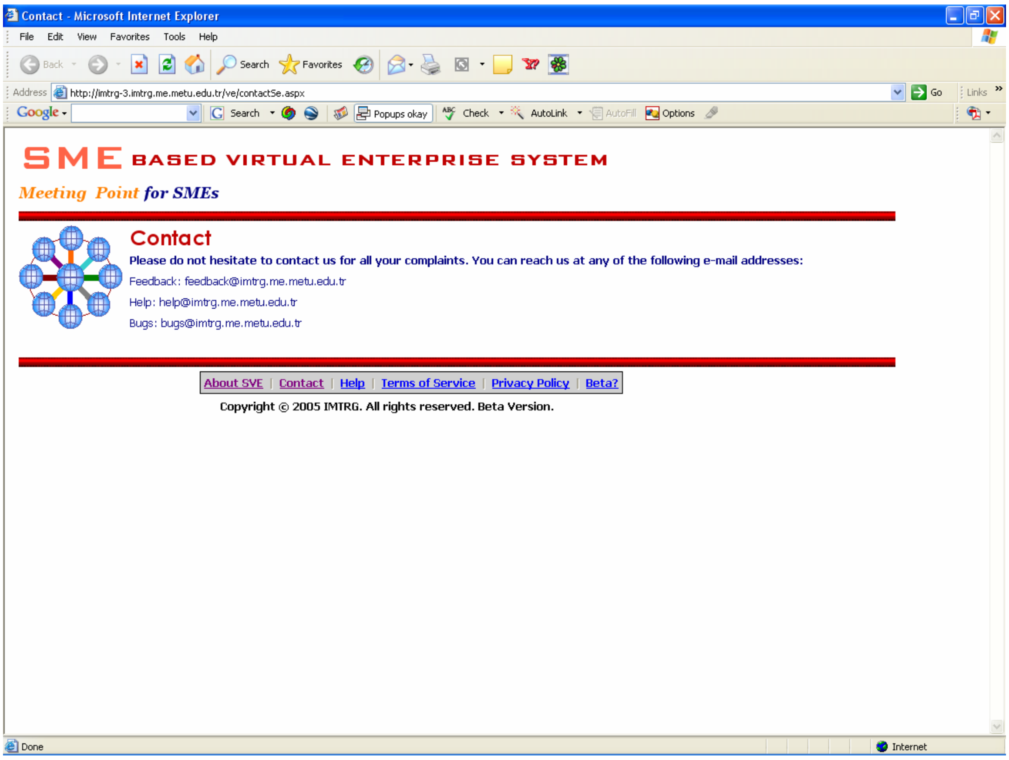


Figure 6.26 Contact page

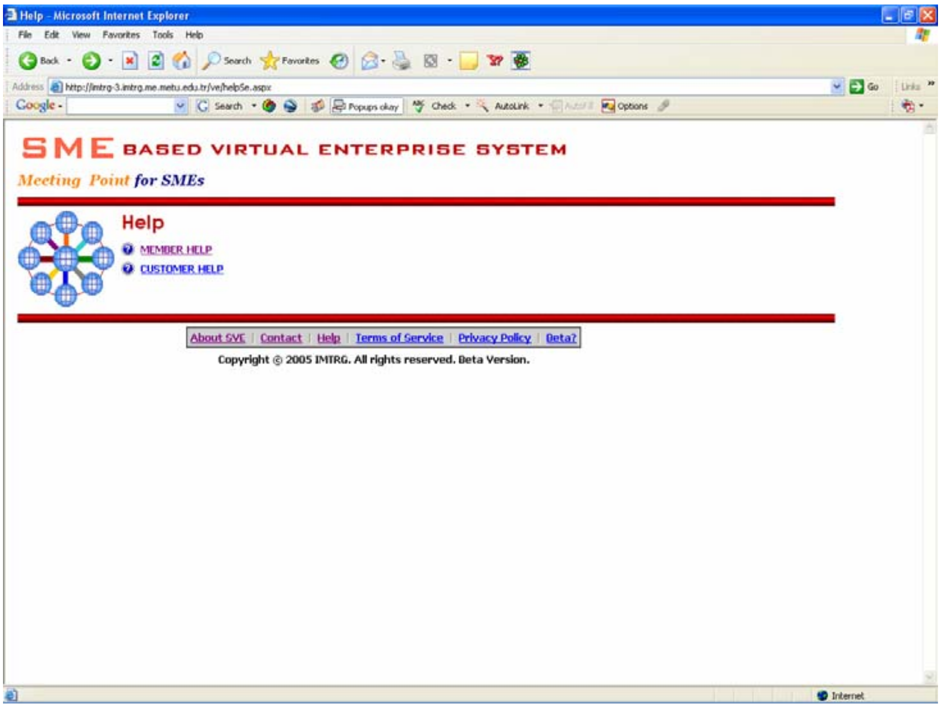


Figure 6.27 Help page

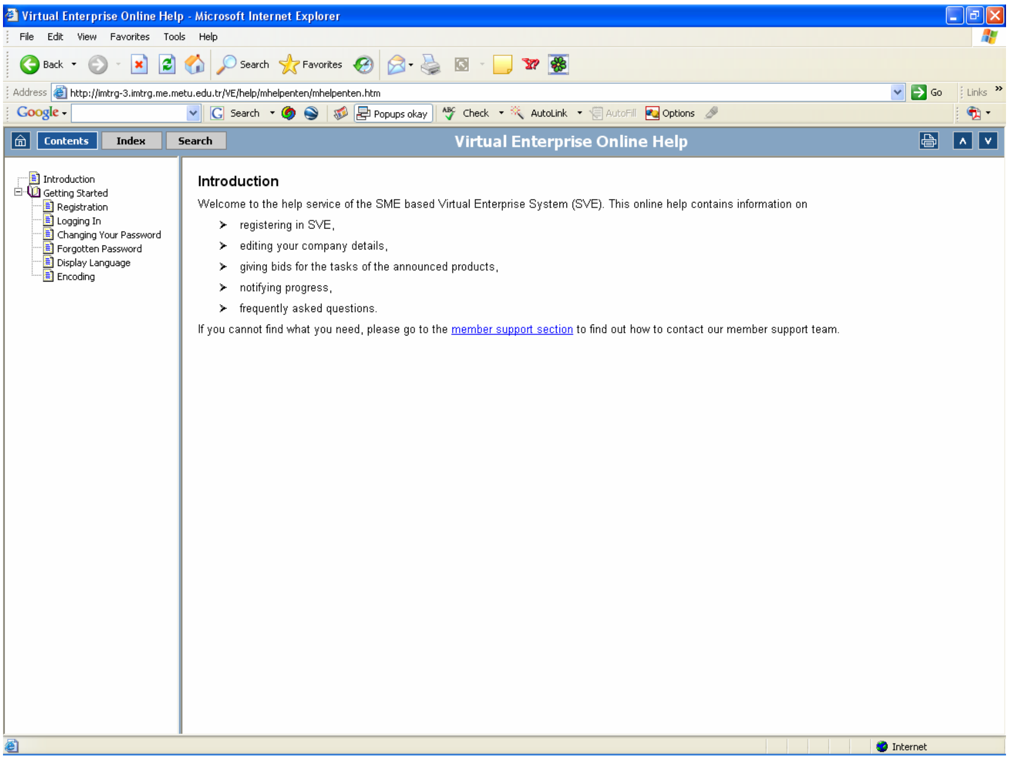


Figure 6.28 Virtual enterprise online help page

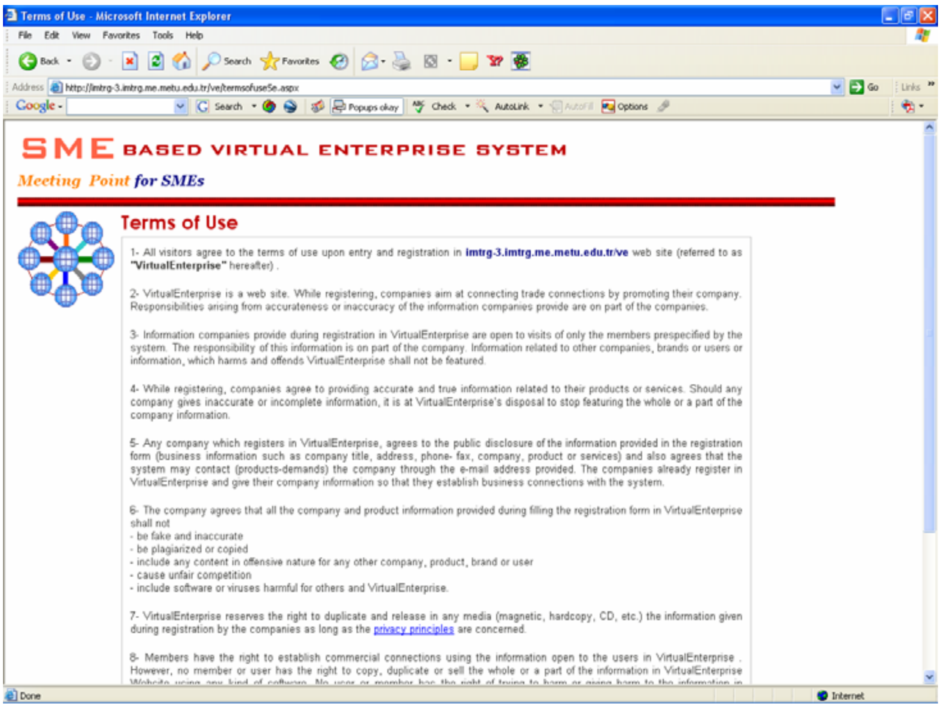


Figure 6.29 Terms of use page

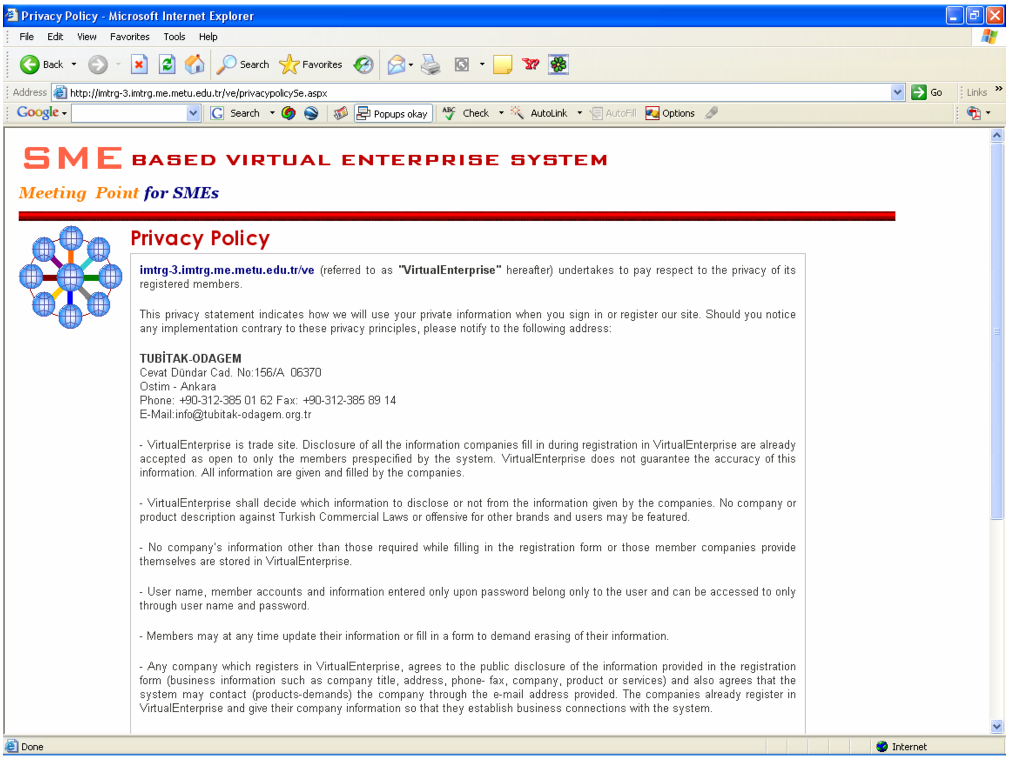


Figure 6.30 Privacy policy page

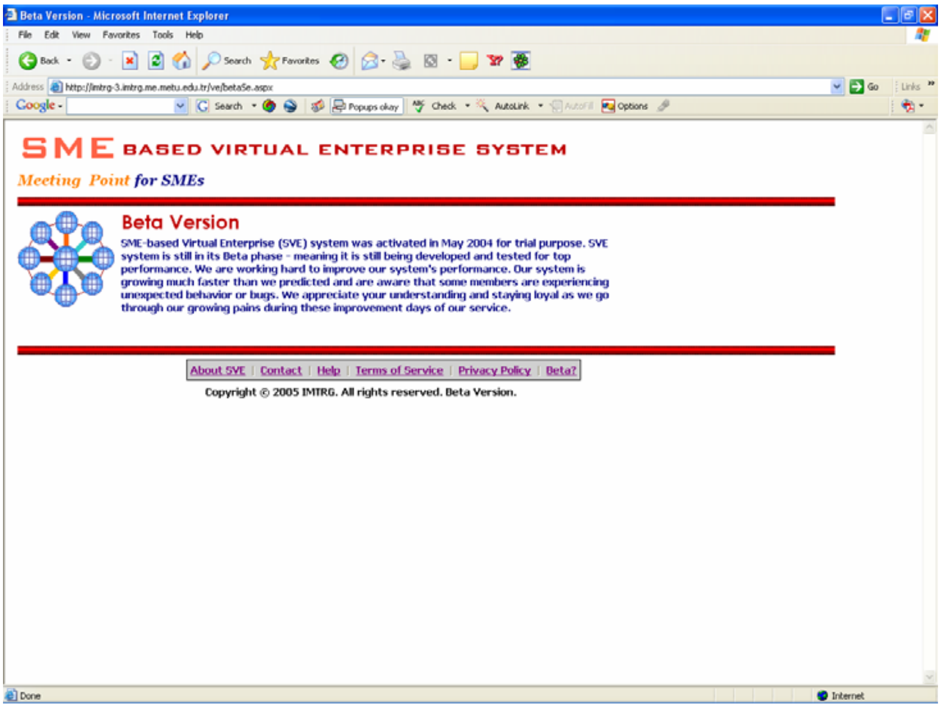


Figure 6.31 Beta page

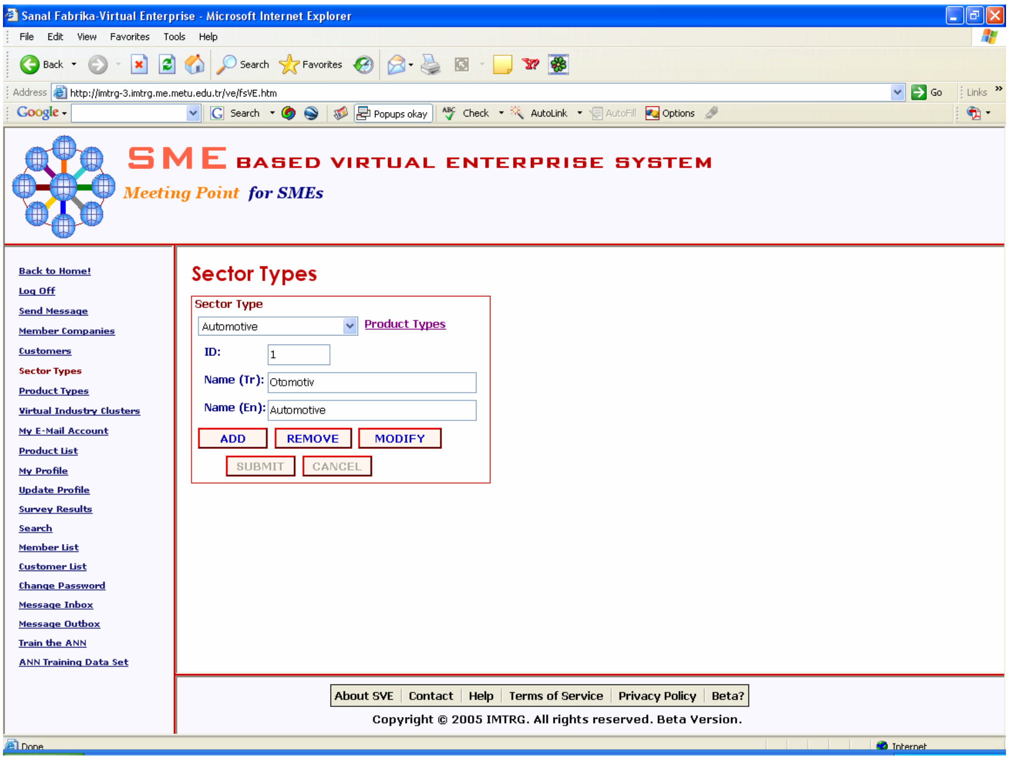


Figure 6.32 Sector types

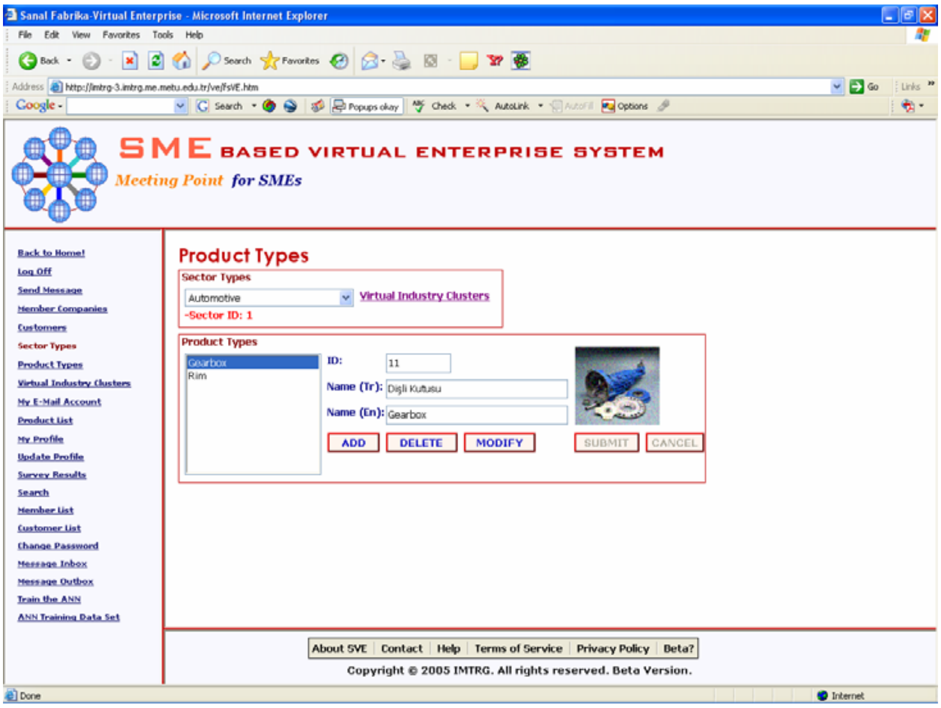


Figure 6.33 Product types page

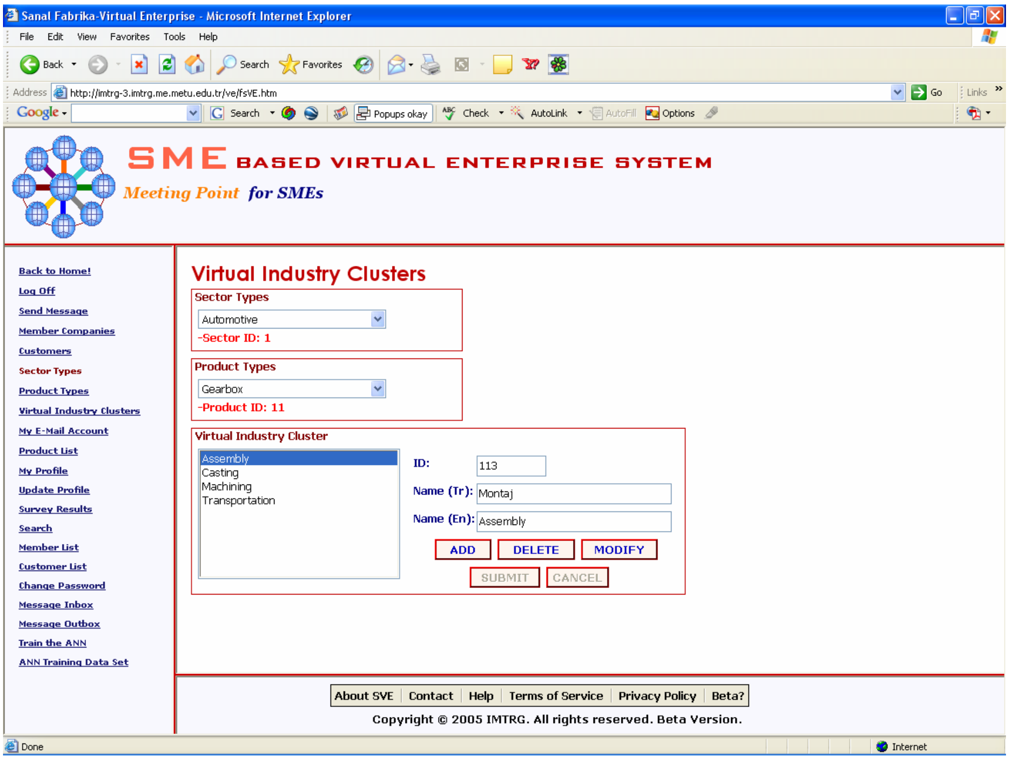


Figure 6.34 Virtual industry clusters page

6.3 Application of AHP & PERT in the VE System

Partner selection depends upon many factors. The factors identified in this research include unit price of the task, caution cost, completion probability of the task and past performance of the partner. These factors act as attributes in AHP and partners considered serve as alternatives thus constituting the last level of hierarchy. The decision hierarchy for the partner selection is shown in Figure 6.35. The factors affecting the partner selection are as follows:

(1) Unit cost: Cost is a major factor which influences the partner selection. A partner bid involving higher cost is liable to be rejected on economic ground. The total task quantity also influences the overall price.

(2) Caution cost: Level of commitment is measured in terms of a caution cost which is the cost that the partner must pay to the VE if the partner decides the give up

before the assigned task is finished and is secured in the form of letter of credit. Thus, the higher this value is, the more preferable for the VE.

(3) Completion probability: The probability of completing the task by the given time period is calculated using PERT. PERT (Program Evaluation Review Technique) is a review technique which is developed to schedule the projects and to cover uncertainty of activity time estimates (Gray & Larson, 2000). PERT uses three time estimate for each activity. Basically, this means that duration of each activity can range from an optimistic time to a pessimistic time, and a weighted average can be computed for each activity. Knowing the weighted average and variances for each task allows the system administrator to compute the probability of meeting pre-specified task duration.

(4) Past performance: Overall performance of the partner company which is computed by taking the average of the partner performances shown during each previously executed task. Performance variability is based on delivery status of the task (late, early or none), quality of the completed task (% acceptable) and submission status of progress reports (late, early or none). A neural network model is developed to establish the performance score of each partner.

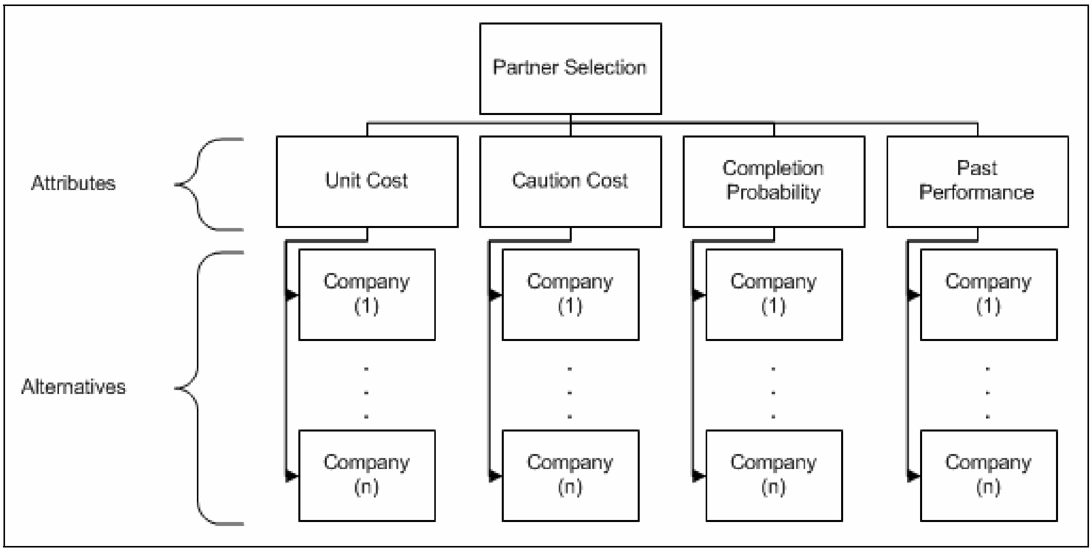


Figure 6.35 Decision hierarchy for the partner selection

6.3.1 Proposed steps of partner selection using AHP

(i) Define the overall objective.

(ii Define the structured hierarchy consisting of attributes (criteria for the partner selection for a given product) and alternatives (member companies).

(iii) Determination of the priority weights of the attributes using pairwise comparison matrix and its consistency ratio.

(iv) Determination of the priority weights of alternatives with respect to attributes (comparison of various company bids with respect to the individual criteria for selection) and consistency ratio for each pairwise comparison matrix.

(v) Enumeration of overall priority weights for all of the alternatives (companies) and consistency ratio for entire hierarchy.

The partner having the highest priority weight will be selected. Various steps of the partner selection process are described in figure 6.36.

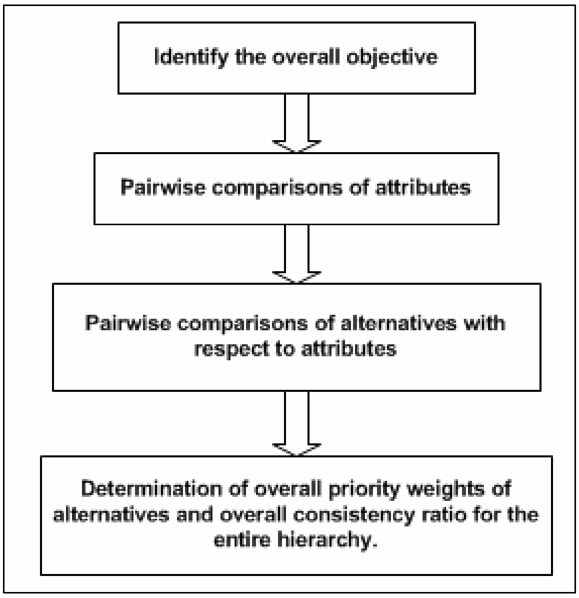


Figure 6.36 Overview of the selection process using AHP

6.3.2 Proposed steps of completion probability calculation using PERT

Knowing the weighted average duration and variances for the task allows the system to compute the probability of meeting given task duration. Reader should follow the steps described in the hypothetical example given next. (The jargon is difficult for those not familiar with statistics, but the process is relatively simple after working through a couple of examples.)

The weighted average task duration is computed by the following formula:

$$T_e = \frac{a + 4m + b}{6} \tag{6.1}$$

where T_e = weighted average task duration

a = optimistic task duration (1 chance in 100 of completing the task earlier under normal conditions)

b = pessimistic task duration (1 chance in 100 of completing the task later under normal conditions)

m = most likely task duration

When three time estimates have been specified, the above equation can be used to compute the weighted average duration for the related task. The variability in the task duration estimates is approximated by the following equations: Equation 6.2 represents the standard deviation for the task.

$$\sigma_{T_e} = \left(\frac{b - a}{6} \right) \tag{6.2}$$

The equation below (Equation 6.3) is used to compute the "Z" value found in the table 6.2 (Z=number of the standard deviations from the mean), which turn, tells the probability of completing the task in the time specified.

$$Z = \left(\frac{T_s - T_e}{\sigma_{t_e}} \right) \tag{6.3}$$

where T_s = scheduled task duration (given as a constraint)

T_e = expected or weighted average task duration

Z = probability (of meeting scheduled duration) found in statistical Table 6.1.

Table 6.2 Z – Probability

Z Value	Probability	Z Value	Probability
-2.0	0.02	+2.0	0.98
-1.5	0.07	+1.5	0.93
-1.0	0.16	+1.0	0.84
-0.7	0.24	+0.7	0.76
-0.5	0.31	+0.5	0.69
-0.3	0.38	+0.3	0.62
-0.1	0.36	+0.1	0.54

6.4 Application of ANN in the VE System

One of the key issues in forming and succeeding the VE is the matter of trust. Trust is also a key requirement in order to make information and knowledge-sharing within these types of strategic alliance work. Partner performance is a trust building element. Therefore, it is important to have some form of performance management to both evaluate individual participation and contribution as well as to score relative performance against the shared goals.

6.4.1 Partner performance model

The first step in the construction of performance model is to determine the factors having effects on the partner performance. The performance of partners is predicted and evaluated under three dimensions:

(1) Quality:

- quality assurance

(2) Delivery:

- observation of deadlines; and
- delivery performance.

(3) Progress:

- observation of deadlines; and
- submission performance

Next step is the determination of the specifications for each assessment factor based on the nature of the problem and formulation of the each factor in terms of its specifications or dimensions.

6.4.1.1 Quality dimension

The "quality assurance" score of the partner is measured in terms of the ratio of the acceptable batch size of the product to the received number of product. One of the conditions for being a member of virtual enterprise is to possess one or more quality system certifications (ISO 9000, plus other relevant scheme). Therefore, assuming that all the partners have at least one of the quality certifications, "quality system" score is not considered in the quality dimension.

$$\text{Quality assurance score} = \frac{\text{Acceptable parts}}{\text{Received parts}} \text{x10} \tag{6.4}$$

6.4.1.2 Delivery dimension

Delivery quantity and delivery delay of a partner should affect his performance, because too less delivery quantity or too late delivery will decrease the productivity. The "observation of deadlines" measures the partner delivery performance with respect to items delivered on time to the partner commit date or contract date.

$$\text{Observation of deadlines score} = \frac{\text{Quantity received on time}}{\text{Quantity ordered}} \text{x10} \tag{6.5}$$

The "delivery performance" measures the number of parts received for an order before or after the quoted delivery date.

$$\text{Delivery performance score} = \frac{\text{Quantity received}}{\text{Quantity ordered}} \text{x10} \qquad (6.6)$$

The below formula calculated the overall delivery score of the partner.

$$\text{Delivery score} = \frac{\text{Observation of deadline} + \text{Delivery performance}}{2} \qquad (6.7)$$

6.4.1.3 Progress dimension

Late or less submission of progress will create distrust between the partner and virtual enterprise. Therefore it should be a critical factor in the evaluation of partner performance. The "observation of deadlines" score is calculated with respect to the number of progress submission on time to the required number of progress submission.

$$\text{Observation of deadlines} = \frac{\text{Progress submission on time}}{\text{Required number of progress submission}} \text{x10} \qquad (6.8)$$

The "submission performance" measures the number of progress submitted for a task.

$$\text{Submission performance} = \frac{\text{Number of progress submitted}}{\text{Required number of progress submission}} \text{x10} \qquad (6.9)$$

The below formula calculated the overall progress score.

$$\text{Progress score} = \frac{\text{Observation of deadlines} + \text{Submission performance}}{2} \qquad (6.10)$$

After the factors affecting the performance value are determined, next step is to construct the neural network for the assessment of partner performance.

6.4.2 Neural network design

Artificial Neural Networks (ANNs) are biologically inspired models analogue to the basic functions of biological neurons. They have a natural propensity for storing experiential knowledge, and resemble the human brain in the sense that training rather than programming is used to acquire knowledge. The capability of learning from the examples probably the most important property of neural networks in applications and can be used to train a neural network with the records of past response of a complex system. Highly complex associations can be directly learned from noisy data without a need to develop mathematical models and expressions.

One of the objectives of this research is to adopt a multilayer neural network to assess the performance of the partner companies for a certain type of business task. Firstly, proper ANN model structures should be defined. There are numerous neural network structures. Each paradigm has its characters and application areas. Most of the neural networks paradigms commonly used have three layers: input layer, output layer, and hidden layer. What should be decided is which ANN model should be used and how many nodes should have in the input layer, hidden layer and output layer.

A neural network consists of a number of nodes massively interconnected through connections. The nodes are arranged in layers: an input layer, an output layer, and several hidden layers. The number of hidden layers depends on the type of problem. The nodes of the input layer receive information as input patterns, and then transform the information through the connections to the other connected nodes layer by layer to the output layer nodes. The transformation behavior of the network depends on the structure of the network and the weights of the connections. Hidden layers act as layers of abstraction, pulling features from the inputs. The architecture of the neurons and connections in the hidden layers depends on the type of problem. Increasing the number of hidden layers will increase both the time and the number of training examples necessary to train the network properly. Decreasing the number of hidden layers will cause an increase in the net error. Unfortunately, the selection of training parameters such as hidden layer is not guided by formal rules or instructions, but has to be performed by testing several different architectures; furthermore, an unsuccessful result requires the change of some parameters, but it does not give any direction (or, at least, very weak directions) to improve the behavior of the network. In general, numerous experiments have shown that backward, multi-layer net with two hidden layers of 2n+1 neurons (where n: number of neurons in input layer) each is appropriate (Zell, 1995). Therefore, the optimal structure of the designed network is summarized as follows: the input layer consists of three neurons, two hidden layers with seven neurons each and the output layer consists of one neuron. If partial knowledge (empirical) exists about the problem, it's worth to integrate it. It is very useful to integrate calculations of existing mathematical models into a network. The designed and fully connected multi-layer neural network is shown in Figure 6.37.

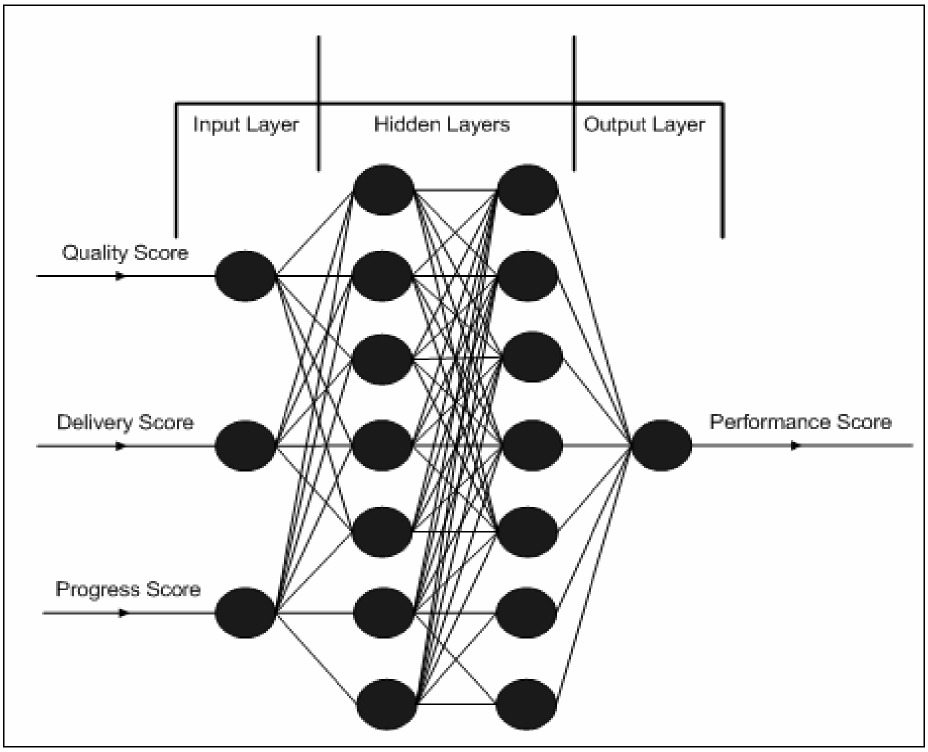


Figure 6.37 Fully connected multi-layered neural network

6.4.3 Training the neural network

Once the neural network has been designed, it has to be trained to produce the expected output values in function of a predefined pattern of input values. This training operation is accomplished by selecting a proper training algorithm for the problem to be solved. Several training algorithms have been developed for ANNs. Many of these training algorithms are closely connected with a certain network topology. Among various existing training algorithms, back propagation algorithm was selected in this research work. It is a widely used algorithm for training of multilayer net for the following reasons: (1) it is most representative and commonly used algorithm and is relatively easy to apply; and (2) it has been proven to be successful in practical applications. The main steps of this algorithm are described in the following subsection.

6.4.3.1 Encoding, decoding and normalizing input data

Data may have to be converted into another form to be meaningful to a neural network. How data are represented and/or translated also plays an important role in the network's ability to grasp the problem, that is, a neural network can learn more easily from some representations than from others. Data may be continuous-valued (e.g. $0 \rightarrow 1$) or binary (e.g. 0, 1). Sometimes data can be represented either as single continuous value or as a set of ranges that are assigned binary representations. When the values are continuous, artificially breaking them up into groups can be a mistake, because it is often difficult for the network to learn examples that have values on or near the border between two groups. When there are naturally occurring groups, the binary categories are often the best method for making correlations. Based on the above mentioned review, the continuous-valued data inputs need to be scaled (normalized) and the binary type inputs needed to be encoded and decoded.

Continuous-valued data must be normalized or scaled if it has a natural range that is different than the network's operating range ($0 \rightarrow 1$). Normalizing is simply dividing all values of a set by an arbitrary reference value, usually the maximum value. Use of the maximum value will limit the maximum value to unity.

To successfully apply neural network in the assessment of partner performance input variables have to be investigated and converted into a suitable format for presentation to the neural network. In this research, the data representations of the neural network inputs are continuous-valued data type and are properly scaled. The value of "10" is used as a scale factor in this work.

6.4.3.2 Back-propagation training algorithm

Back-propagation algorithm is a gradient decent method to minimize the total sum of square error over the entire training data set. The convergence to the optimal solution is accomplished by adjusting the weight connections through the partial derivative of the sum-squared error with respect to the weights. The implementation of this algorithm for training the designed neural network is summarized by the following steps:

Step 1

Set all the necessary network parameters such as the number of input neurons, the number of hidden layers and the number of neurons included in each hidden layer, the number of output neurons, etc.

Step 2

Set all network weights to small random values, positive and negative (-0.3 to 0.3).

Step 3

Initialize the iteration (epoch) number (m=1) and presentation (example) number (n=1).

Step 4

Apply one training sample to the input layer $[X_1, X_2, ..., X_{N_k}]$ and note the corresponding desired output $[O_1, O_2, ...O_{N_k}]$, where N_k is the number of neurons in layer k.

Step 5

Calculate the output of the neurons layer by layer through the network, from the second layer to the output layer using:

$$O_j^{N_k}(n) = F\left(\sum_{i=1}^{p} W_{ji}(m) O_i^{N_{k-1}}(n)\right) \tag{6.11}$$

for each neuron j, $1 \leq j \leq N_k$ and $2 \leq k \leq L$

where L is the number of layers in the network.

F is a sigmoid activation function of the form $F(a) = 1/(1+e^{-a})$.

Wji(m) is the weight connecting neuron i in layer k to neuron j in layer k+1.

$O_i^{N_k}$ is the output of neuron N in layer k.

Step 6

Calculate the error gradient δ for every neuron in every layer in backward order from output to the first hidden layer.

The error for the output layer neurons is computed by

$$\delta_j^{N_k}(n) = O_j^{N_k}(n)\left(1 - O_j^{N_k}(n)\right)\left(T_j(n) - O_j^{N_k}(n)\right) \tag{6.12}$$

for every neuron j, $1 \leq j \leq N_k$, k=L where $T_j(n)$ is the target vector. Then, successively, the error gradients for all hidden layer neuron are computed from;

$$\delta_j^{N_k}(n) = O_j^{N_k}(n)\left(1 - O_j^{N_k}(n)\right)\sum_{i=1}^{N_{k+1}} \delta_i^{N_{k+1}}(n) W_{ij}(m) \tag{6.13}$$

for every neuron j, $1 \le j \le N_k$, k=L-1,……,2

At the end of the error backward propagation step, neurons of the network will have error values (except input layer neurons, L=1).

Step 7

Adjust the network weights for every layer. Starting at the output layer neurons and working back to the first hidden layer recursively adjust weights according to the generalized delta rule.

$$W_{ji}(m+1) = W_{ji}(m) + \eta \delta_j^{N_k}(n) O_j^{N_k}(n) + \alpha \left[W_{ji}(m) - W_{ji}(m-1) \right] \quad (6.14)$$

for every neuron j, $1 \le j \le N_k$, k=L,L-1,…,2

where α is momentum constant (0< α<1) to smooth out the weight change and accelerate convergence of the network.

η is learning rate (0< η<1) controls the step size for weight adjustments.

Step 8

Repeat actions in steps 4 to 7 for every training sample.

Step 9

Calculate the average sum-squared error resulted at the end of every training cycle. This error can be evaluated by the following expression.

$$sse = \frac{1}{2N} \sum_{j=1}^{n} \sum_{i=1}^{N_L} (T_{ij} - O_{ij}^L)^2 \quad (6.15)$$

where Tij is the target value desired for the i^{th} output and for the j^{th} example.

Step 10

Compare the average sum-squared error (sse) with the predetermined limit value (tolerance value) (ε) of the error, if it is less then stop. Otherwise, increase number of iterations and randomize the order in the training set and return to step 4.

6.4.3.3 Training data patterns

A successful neural network requires that the training data set and training procedure be appropriate to the problem. The selection of the training parameters takes place only on the basis of the programmer's experience. There are no general rules for their selection. The training data set must span the total range of input patterns sufficiently well so that the trained network can generalize about the data. In order to have extrapolation and interpolation capabilities, neural networks must be trained on a wide enough set of input data to generalize from their training sets. To achieve this goal and demonstrate the applicability of the designed neural network, 374 training patterns (each pattern is formed by input and output vectors) are used to generate the performance value. The input values of the training patterns are selected from within specified range for each input parameter. The output values are based upon the limitations put on each variable.

Training data patterns can be obtained from the data handbooks, experiments, interviews and case oriented programs generating required output. Several data handbooks do not guarantee the best or the most appropriate, but only "good". The reason is that they are grounded on the experimental results as well as obtained by considering only limited number of operating factors. Data that was gained through experience is the most valuable one. Table 6.3 presents selected samples of these patterns (Sarkis & Talluri, 2002). Note that Score Point (SP) ranging from 0 (least score) to 10 (highest score) is used to assess partner company's performance.

6.4.3.4 Training and testing experiments

Several training experiments have been carried out to identify the optimal network structure and best training parameters of the neural networks which produces minimum errors during training phase (Sari et al, 2006b). The adopted procedure is schematically represented in Figure 6.38; after setting a meaningful set of examples to be initially used in training, a sample structure with a low number of neurons has been firstly considered; during the process, the complexity of the structure has been progressively increased until the network error reaches a value lower than a predefined limit. In this way, several training experiments with a different number of hidden neurons, learning rate, and momentum values have been checked $\leq\varepsilon$.

Table 6.3 Selected samples of training patterns (adapted from Sarkis, 2002)

Input vector			**Output vector**
Delivery	Quality	Progress	Performance
SP	SP	SP	SP
8	8	2	6.8
7	8.5	3.3	6.86
9	8.3	4	7.72
9.2	8.1	6	8.12
3	10	7.5	6.7
10	9	10	9.6

In particular, a given network structure has been considered able to correctly learn the training set if the following condition is satisfied:

$$sse \leq \varepsilon \tag{6.16}$$

where ε is the predefined limit value of the error (tolerance value) of the error and sse is the average sum square error resulted at the end of training process. During the training process, the error limit has been set to 0.00001.

At the end of training phase, the generalization capability of the network is verified by a validation test presenting pairs of input and target vectors (test sample) describing intermediate situations with respect to those proposed in training. If the test sample produces an acceptable result then the neural network is validated; if the result is unacceptable, the following corrective actions are considered:

- Reduction of the network complexity and changing the training parameters, supposing that this behavior is caused by over-fitting phenomena.
- Increase of number of proposed training examples, adding to the training set the situations of the test sample which have generated unacceptable results.

In this research work, the second option has been adopted. It is obvious that modification of the training set requires performing again the whole procedure of training and testing. The optimal structure of the designed network, which gave minimum error, is summarized as follows: the input layer consists of three neurons, two hidden layers with seven neurons each and the output layer consists of one neuron. The learning parameters are 0.9 for the momentum and 0.93 for the learning rate. Momentum is used to increase the rate of convergence by preventing the search from falling into shallow local minima during the search process. Learning rate controls the step size when weights are iteratively adjusted. An average error of 9.88×10^{-5} has been reached after the 374 training samples (selected for training the designed network) have been presented to the network 9516 times. Figure 6.39 shows the training progress of the network. The graph illustrates downward movement of the error rate as learning progressed, indicating that the average error decreased between actual and predicted results.

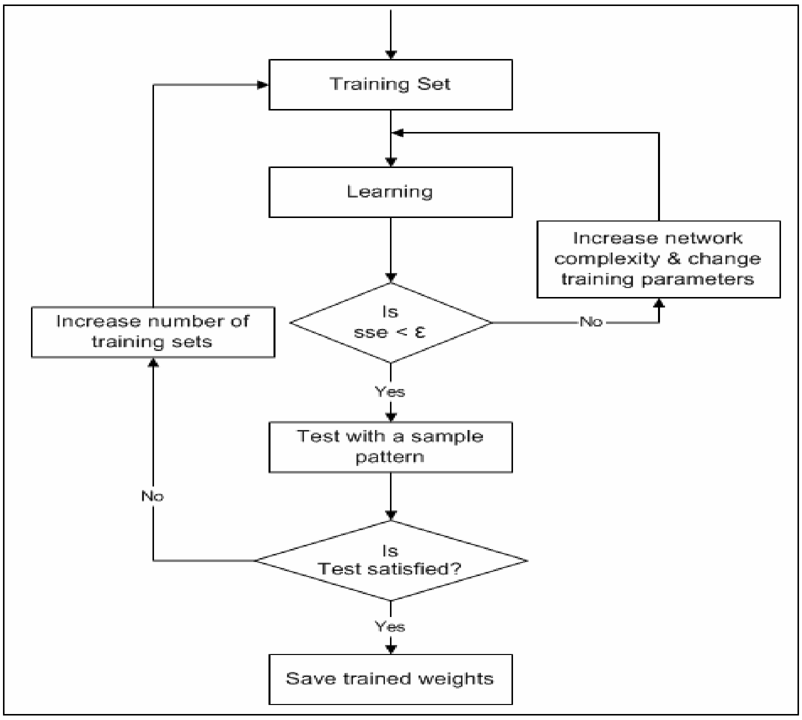


Figure 6.38 Training and testing procedure of neural network

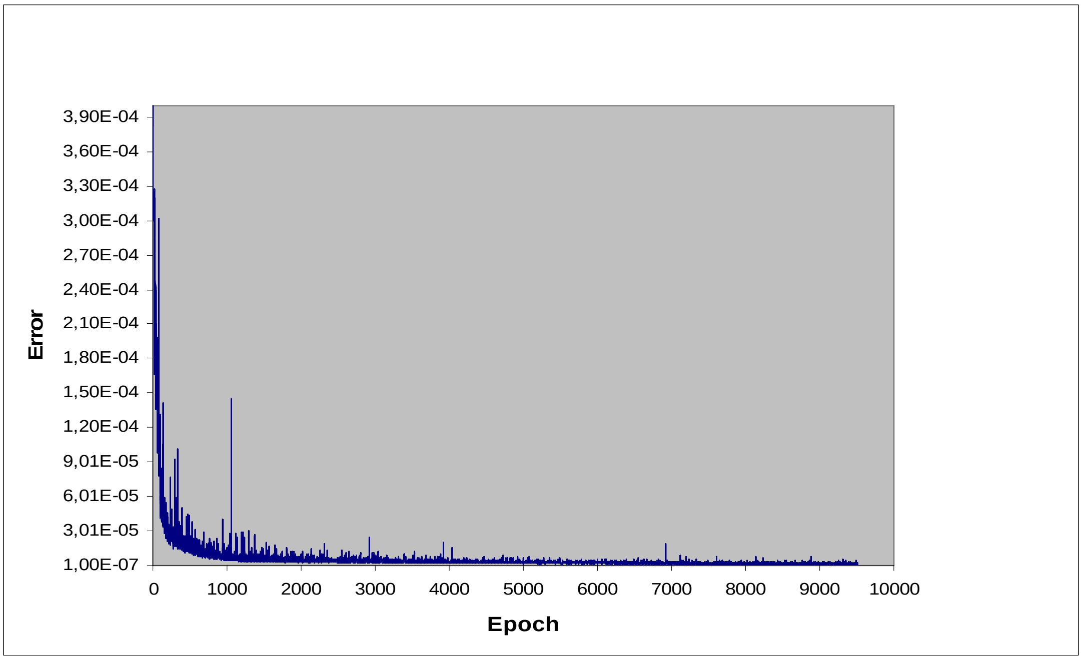


Figure 6.39 Training progress of the network

CHAPTER 7

VIRTUAL CASE STUDY & VALIDATION PLATFORM

The aim of this chapter is to address the fourth research question: *RQ4: How to test & validate VEM?* One of the aims of this research is to carry out certain test cases in order to test the proposed VE methodology. The validation of the system has been done by the following activities; meetings, conferences, presentations and publication of journals.

7.1 Virtual Case Study

To demonstrate the application of the developed VE methodology, a virtual case study entitled as "production of a ball valve for general use" is presented in this section to illustrate the key activities related to setting up breeding environment, setting up & operating VE and dissolution of VE.

Ball valves are quarter-turn, straight through flow valves that have a round closure element with matching rounded seats that permit uniform sealing stress. Ball valves are used in situations where tight shut-off is required. The type of seat can vary with the valve pressure rating and materials of construction. Some valve seats are composed of single molded forms, while other seats with higher-pressure ratings often incorporate a design where each face of the ball is separately sealed. They are wide duty valves, able to transfer gases, liquids and liquids with suspended solids (slurries). Sample view of a ball valve for general use is given Figure 7.1.

Ball valves consist of a rotating ball with a hole in the center that controls straight through flow when the valve is in the open position. They are used in situations where tight shut-off is required.

Figure 7.1 Ball valve for general use

Application fields of valves for general use can be listed as follows: water, hot water, non-acidic liquid fluids, pressure air, non-combustible and non-explosive gas fluids. Valves for general use consists of various types of parts as shown in Figure 7.2. Table 7.1 lists out the part names, materials and production processes.

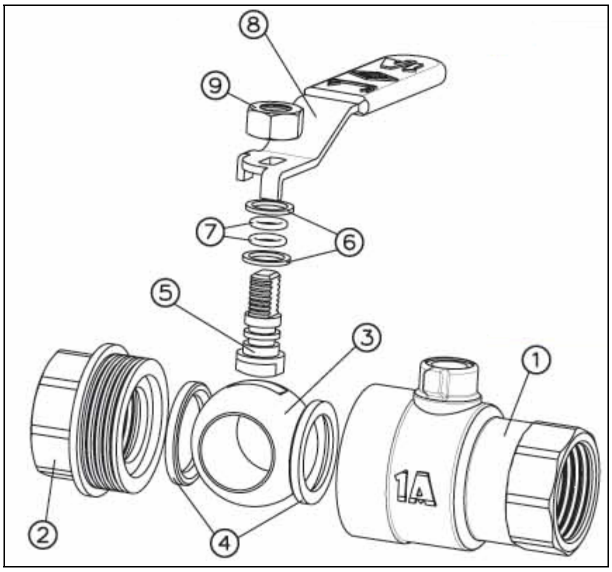


Figure 7.2 Assembly view of ball valve for general use

Generally, valves for general use are called according to the nominal size. Mostly used dimensions of the ball valve are shown in Figure 7.3. In this case study, we are assuming that valve for general use with a nominal size of DN20 is being ordered. In table 7.2, mostly used dimension values for DN20 are listed.

Table 7.1 Part list of ball valve for general use

	Part Names	Material	Production Processes
1	Body	Brass	Hot forging (Ra: 6.3 μm), sand blasting, machining (Ra: 3.2 μm), nickel coating (thickness: 5 μm)
2	Cap	Brass	Hot forging (Ra: 6.3 μm), sand blasting, machining (Ra: 3.2 μm), nickel coating (thickness: 5 μm)
3	Ball	Brass	Hot forging (Ra: 6.3 μm), sand blasting, machining (Ra: 3.2 μm), polishing, nickel coating (thickness: 5 μm) , chromium coating (thickness: 2 μm)
4	Seat	Virgin PTFE	Hardware supplier
5	Stem	Brass	Machining (Ra: 3.2 μm), nickel coating (thickness: 5 μm)
6	Washer	Virgin PTFE	Hardware supplier
7	O-Ring	Viton	Hardware supplier
8	Lever	Steel + PE	Press, vibration, galvanized coating, plastic coating
9	Nut	Steel	Hardware supplier

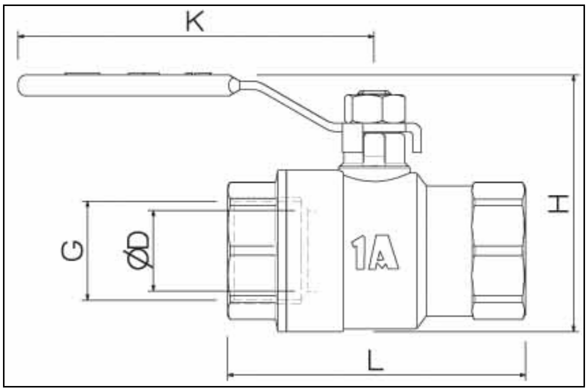


Figure 7.3 Dimensioning ball valve for general use

Table 7.2 Dimensions for DN20 ball valve

Nominal size	G	D (mm)	L (mm)	K (mm)	H (mm)	Kvs(m3/h)
DN20	G 3/4"	19,2	70	85	61	22

7.1.1 Setting up breeding environment

Key activities related to setting up a network for “valve for general use” are given in below. These activities, of course, are not independent of other network entity phase activities.

7.1.1.1 Determine VBE Mission

In this stage, VE coordinator should define the product type including the type of sector it shall address and consequently which types of virtual industry clusters are required to fulfill the identified needs. Sample views related with the determined VBE mission are shown on Figures 7.4 to 7.6.

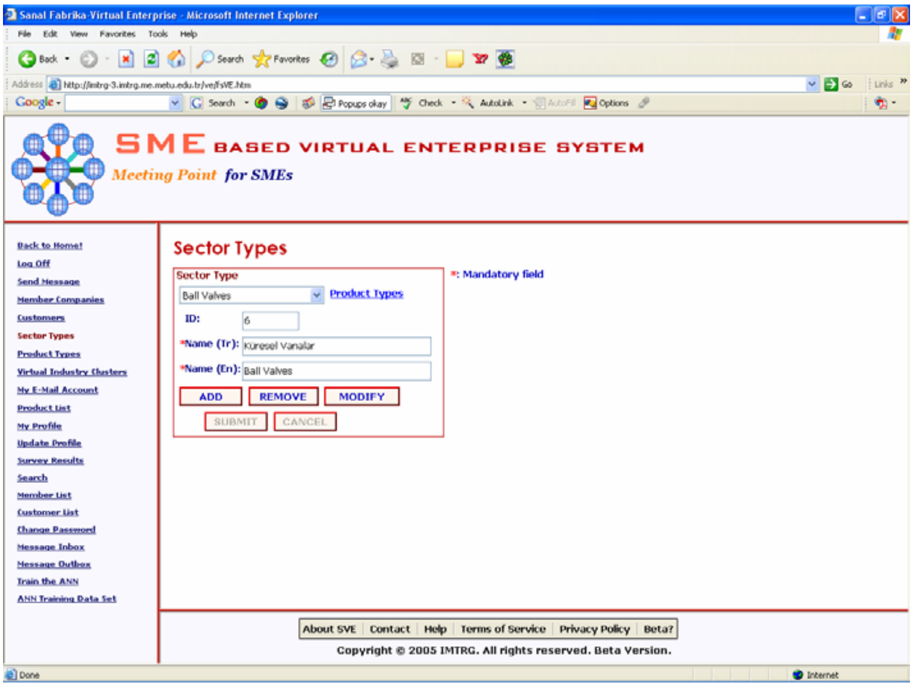


Figure 7.4 Sector type page

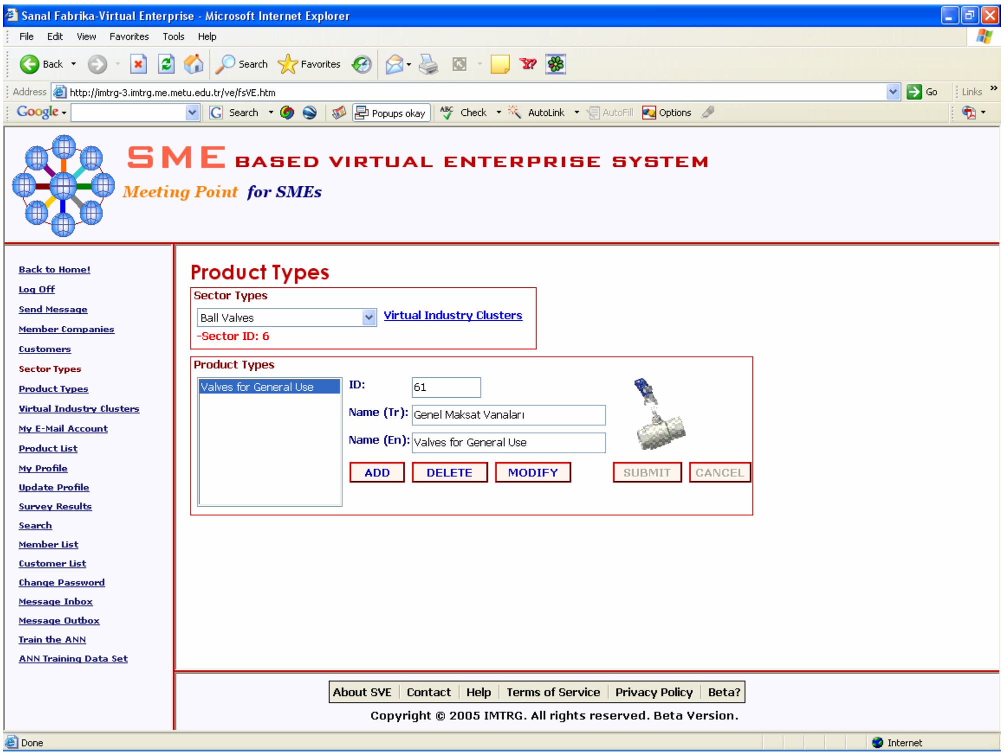


Figure 7.5 Product type page

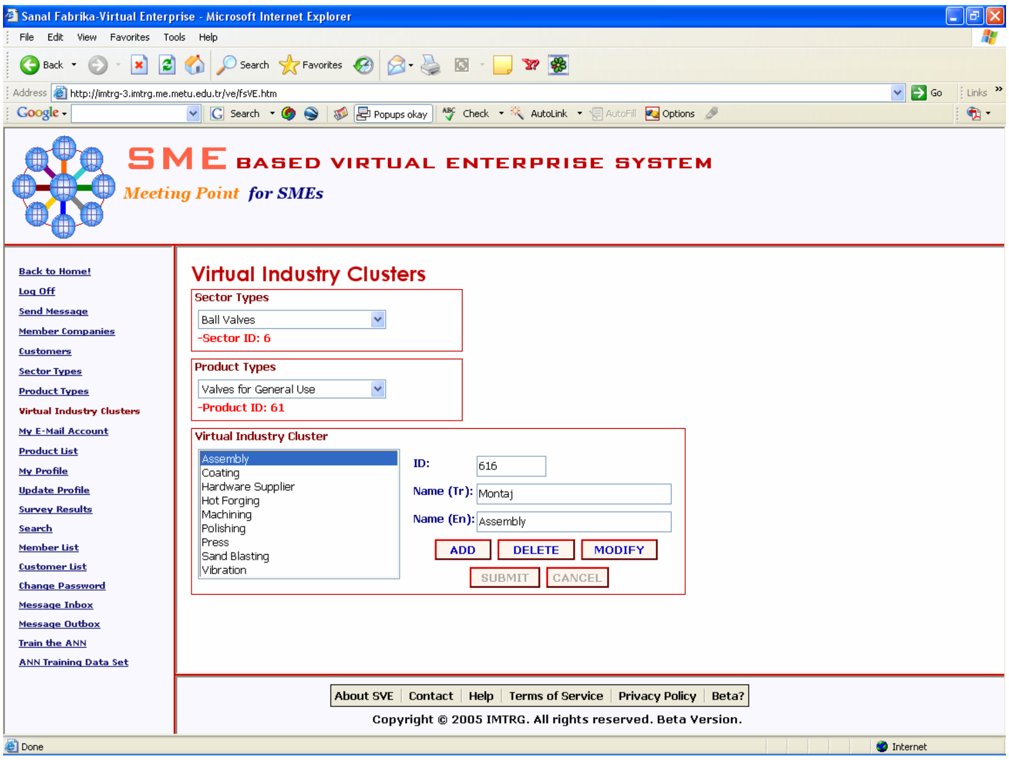


Figure 7.6 Virtual industry cluster page

7.1.1.2 Registration of interested enterprises

Sample views related with the registration stage are given on Figures 7.7 to 7.8.

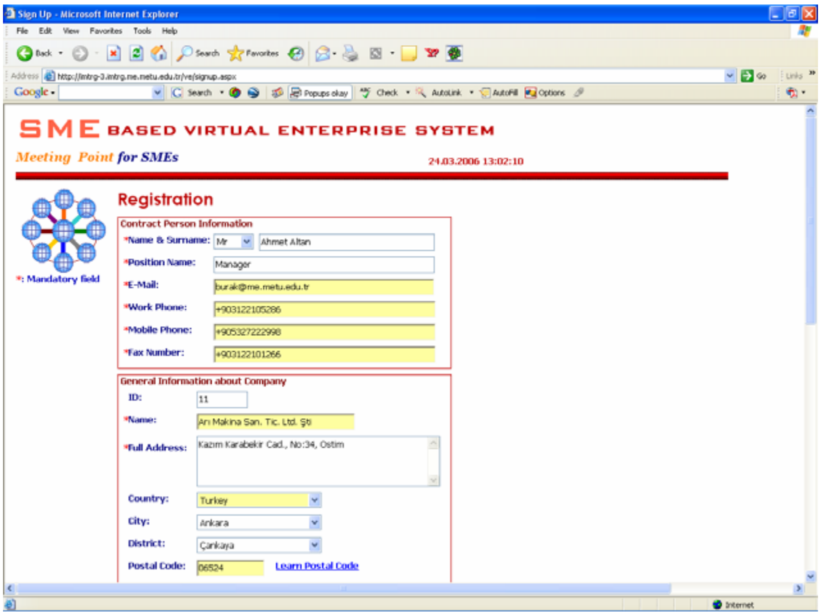


Figure 7.7 Registration of interested enterprise

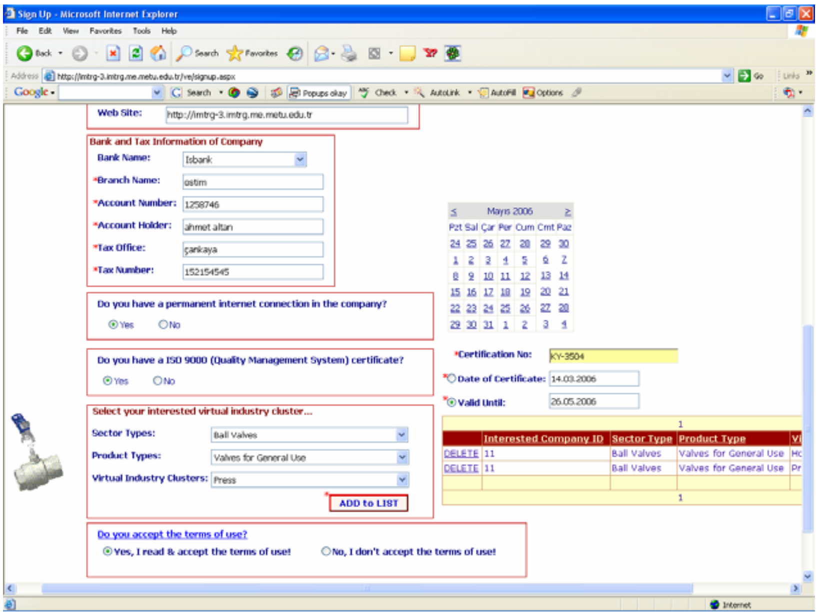


Figure 7.8 Registration of interested enterprise – continued

In this stage, interested enterprises should enter the contact person information, company details, bank & tax information of the company and specify the virtual industry cluster(s) they are interested in to join. In summary, they are required to enter data in those fields marked with a red asterisk (*), otherwise registration can not be completed successfully. They are also asked to answer questions that related with the company infrastructures. Once the interested enterprises have submitted all the required registration information, they will receive an e-mail that gives information about the evaluation process.

7.1.1.3 Select VBE Member

Sample views from the evaluation procedure are shown on Figures 7.9 to 7.12.

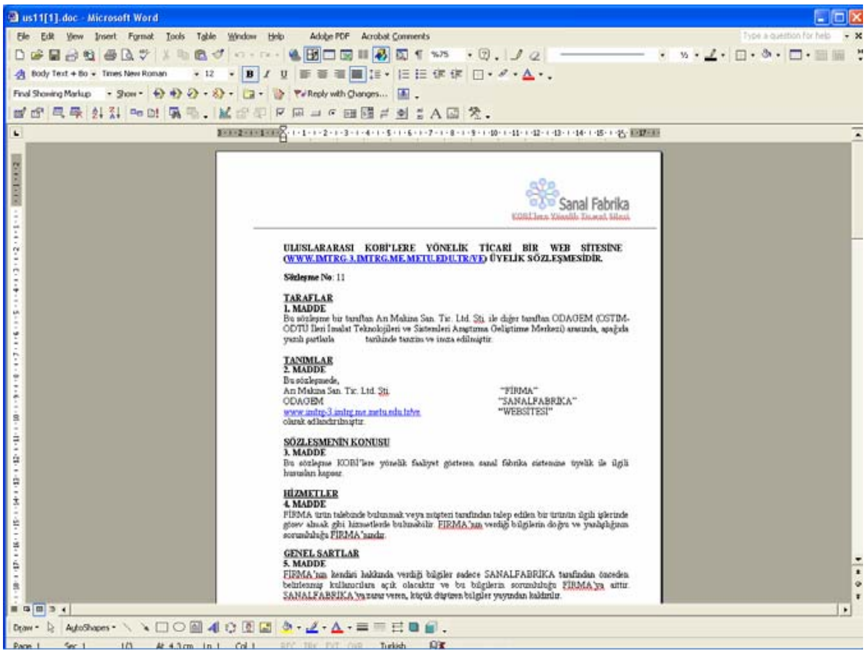


Figure 7.9 Membership contract for interested enterprise

During this evaluation process, VE coordinator will visit the interested company to verify the information given in the registration process and match the core competencies, cooperation agreements and infrastructures of the interested enterprise against the VBE requirements which are listed below:

- Terms of use should be read & accepted by the interested enterprise,
- Interested enterprise should have a permanent internet connection,
- Internet enterprise should have ISO 9000 certificate.

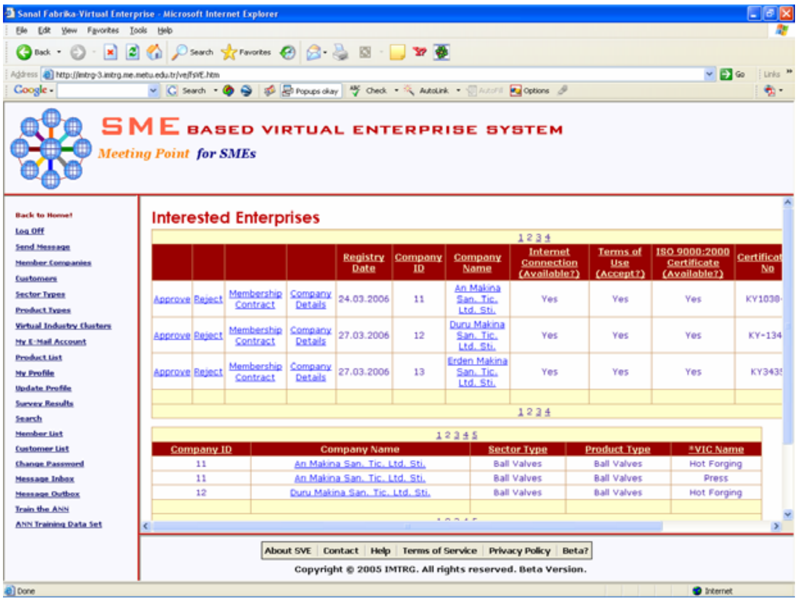


Figure 7.10 List of interested enterprises

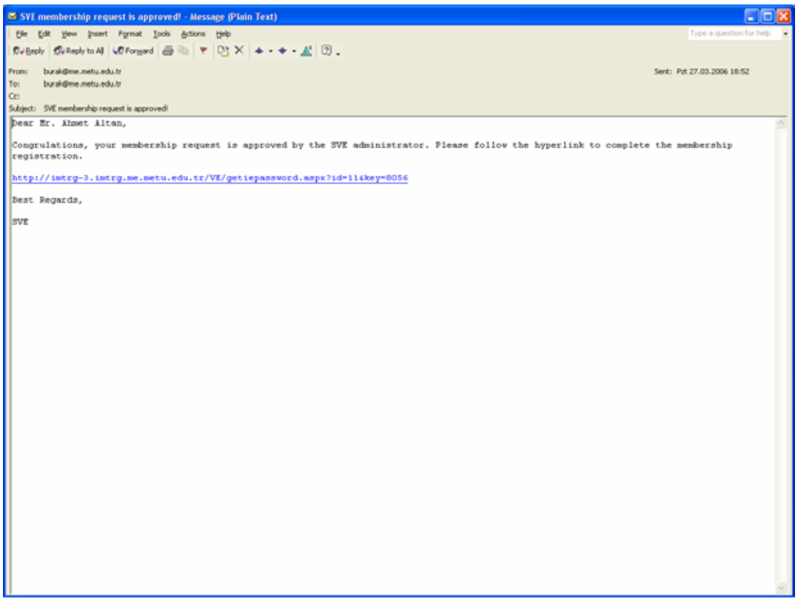


Figure 7.11 Notification e-mail to interested enterprise

If VE coordinator finds the company suitable to become a member of the virtual enterprise, then a membership contract will be signed between the VE coordinator and the interested enterprise.

Once the membership application has been accepted by the coordinator, interested enterprise will receive an e-mail in which information about how to get a username and password will be attached. Following the hyperlink given in the e-mail, the interested enterprise will be redirected to a web page where he/she should select a username and a password. A secret question and an answer to this specific question should also be provided. User details (user name, password, secret question and answer) will be sent to the interested enterprise via e-mail after the submission of the related data.

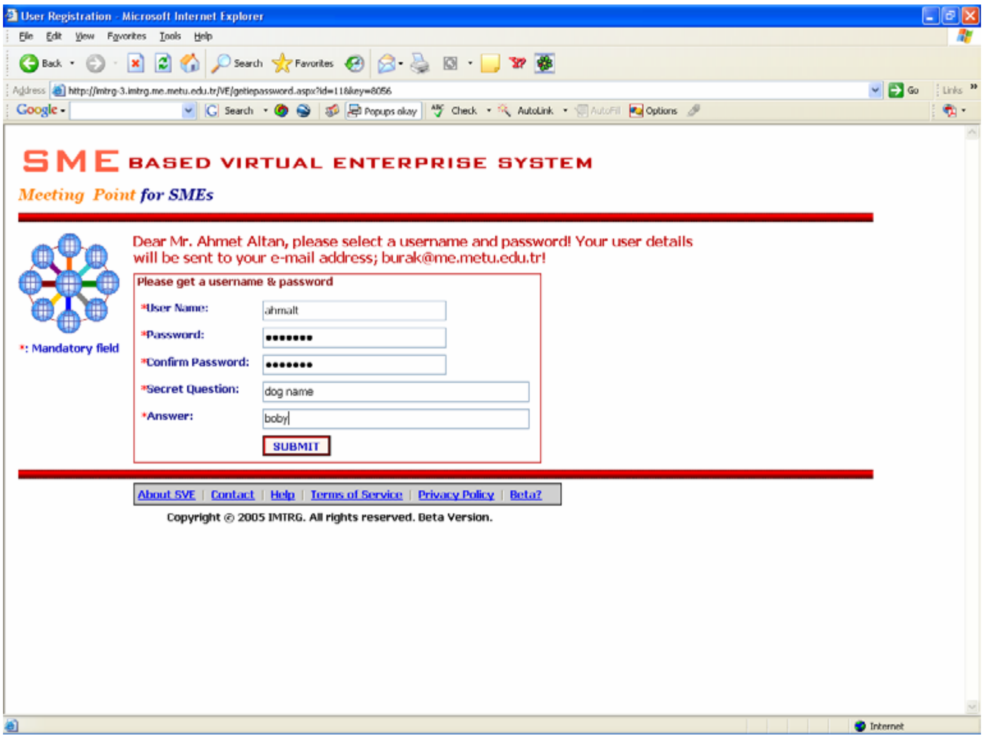


Figure 7.12 User details page

7.1.2 Setting up virtual enterprise

Within the project steps, while all the members' competencies are being identified and the marketing campaign starting, VEs can be set up in two situations: a customer can request a product or a member of the cluster can come up with a business opportunity that he could not explore by himself. Key activities related to setting up a VE for "valve for general use" are given in below.

7.1.2.1 Registration of customer candidates

Sample views related with the registration phase are given on Figures 7.13 to 7.14.

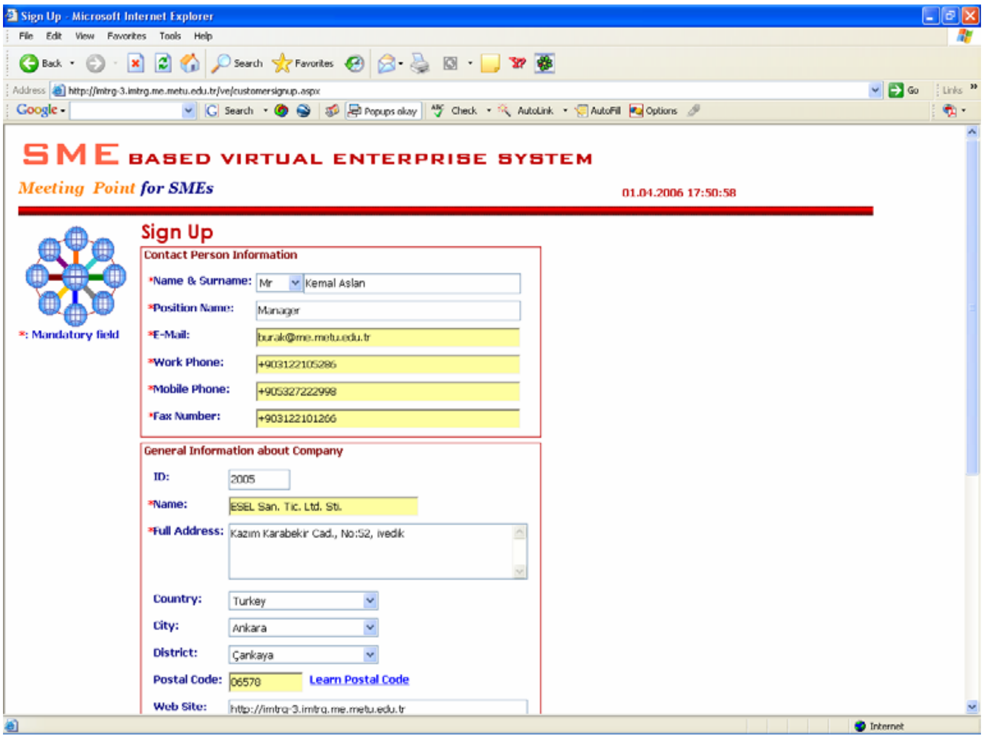


Figure 7.13 Registration of a customer candidate

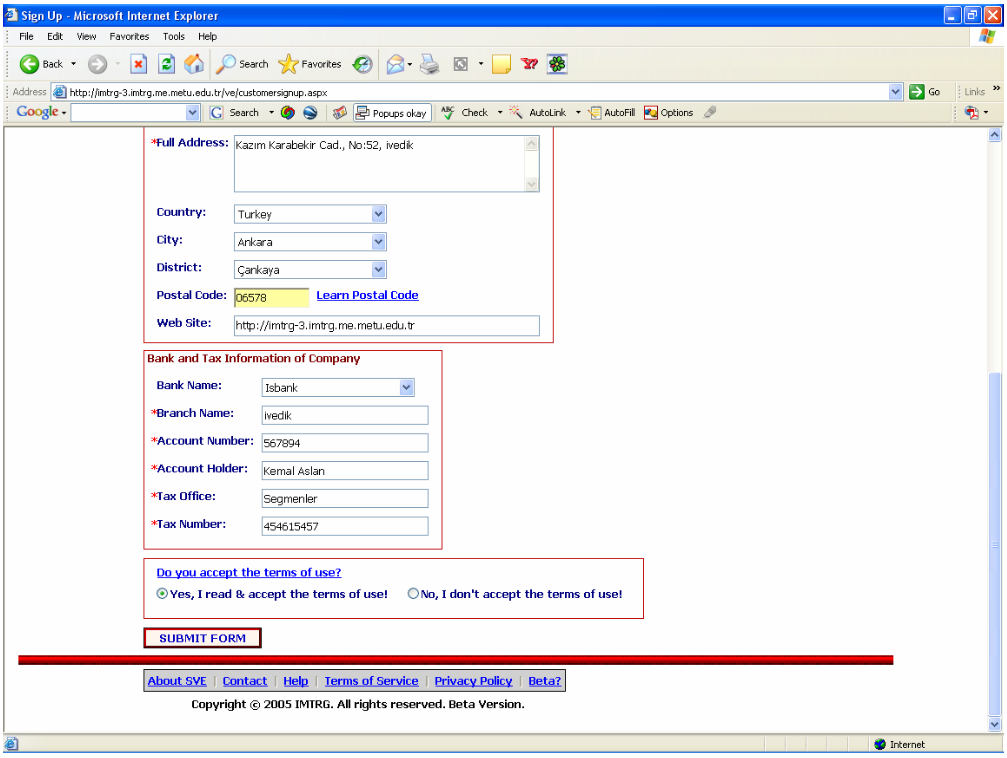


Figure 7.14 Registration of a customer candidate – continued

In this stage, a customer candidate willing to give a product demand should first register to the system similar to the interested enterprises.

7.1.2.2 Select customer

Sample views from the selection procedure of customer are shown on Figures 7.15 to 7.17.

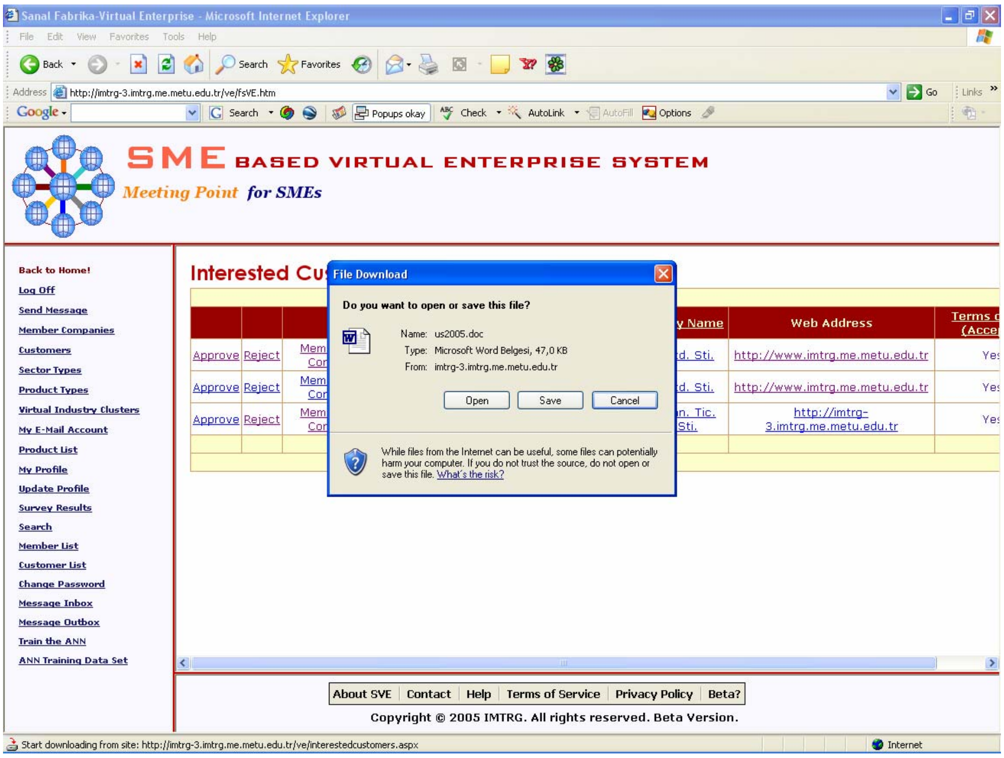


Figure 7.15 Membership contract for customer candidate

After the customer candidate is registered for membership, VE coordinator should visit the customer and validate the given data. After the validation of the customer information, a membership contract should be signed between the coordinator and the customer candidate to formalize the business relation.

Once the membership application has been accepted by the coordinator, the customer candidate will receive an e-mail in which information about how to get a username and password will be sent. Following the hyperlink given in the e-mail, the customer candidate will be redirected to a web page where he/she should select a username and a password. A secret question and an answer to this specific question should also be provided. User details (user name, password, secret question and answer) will be sent to the customer via e-mail after the submission of related data.

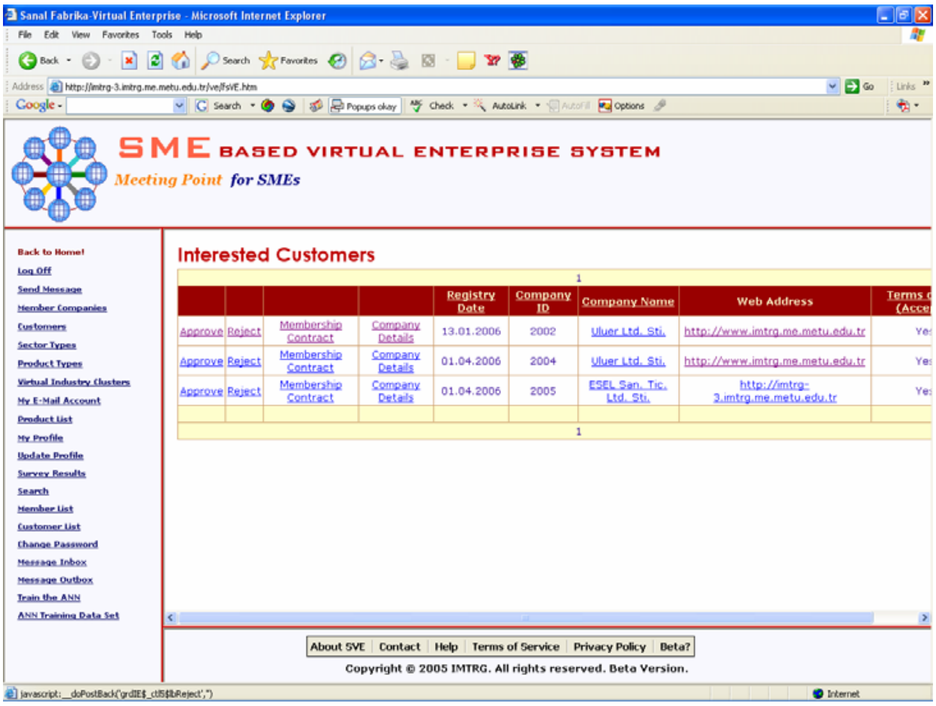


Figure 7.16 List of interested customers

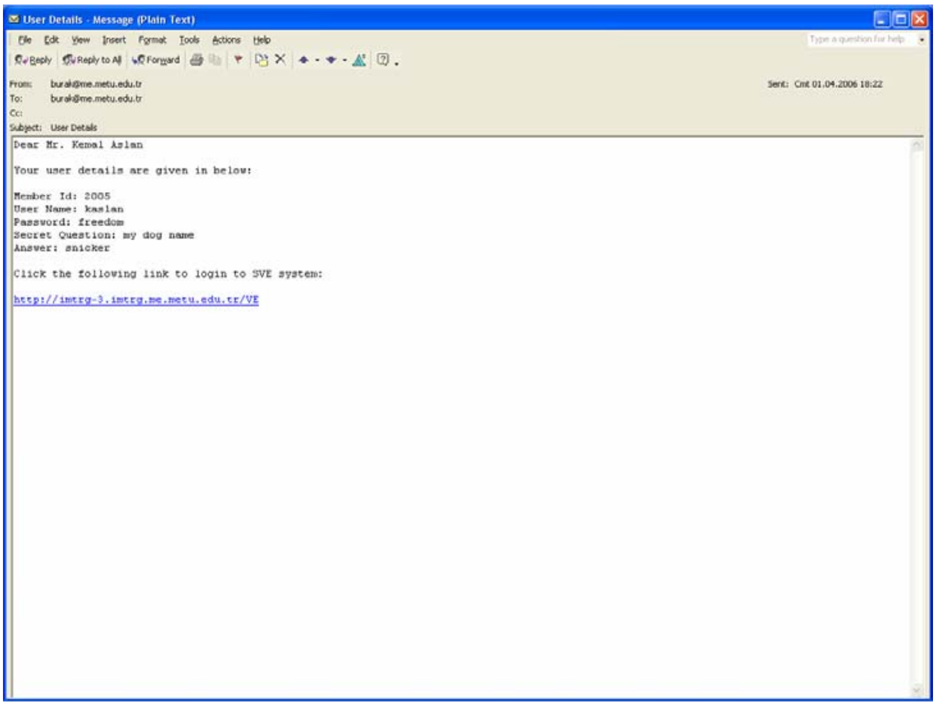


Figure 7.17 Notification e-mail including user details

7.1.2.3 Product demand

Sample views from the product demand are shown on Figures 7.18 to 7.20.

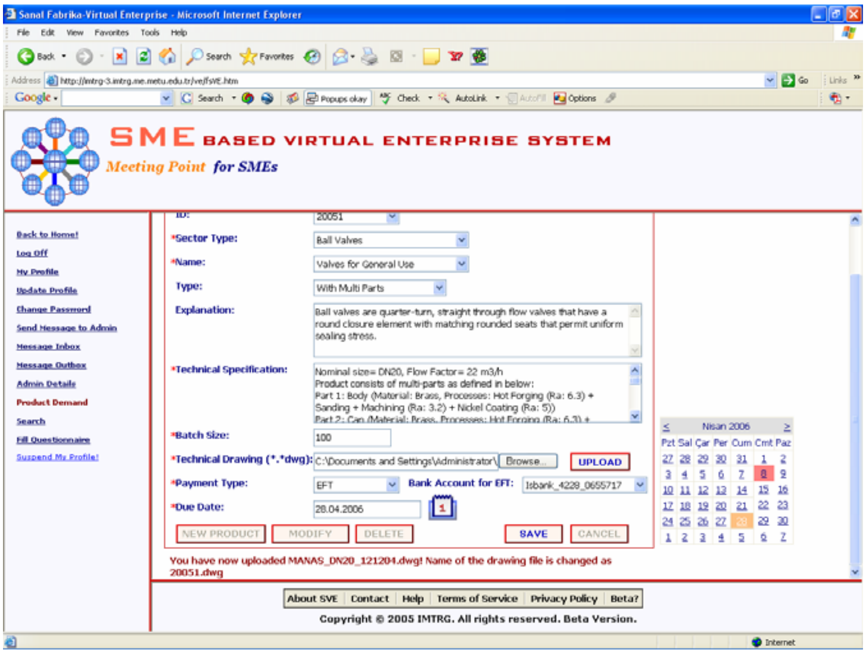


Figure 7.18 Product demand page

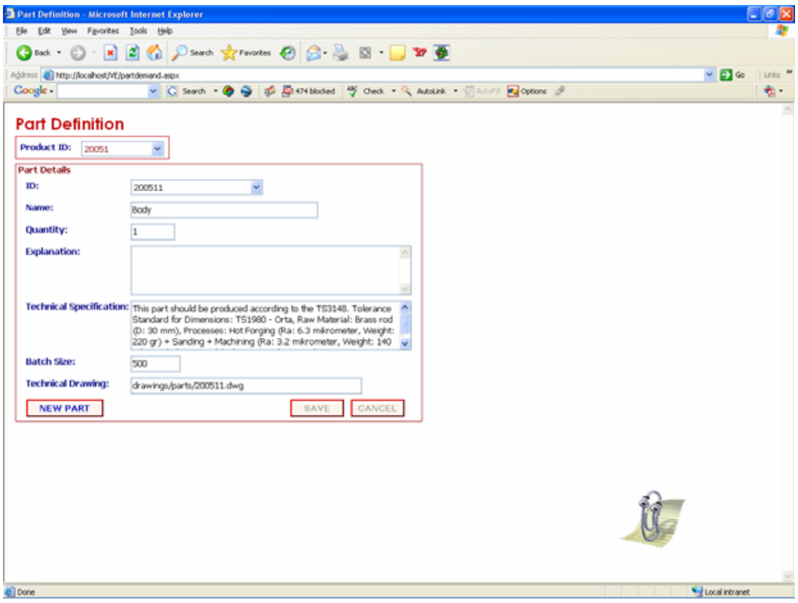


Figure 7.19 Part definition page

To demand a new product, the customer is required to submit the product information. Once the required information is given completely, the customer will be redirected to the Part page where he/she can define the parts of the product one by one if the product consists of multi parts.

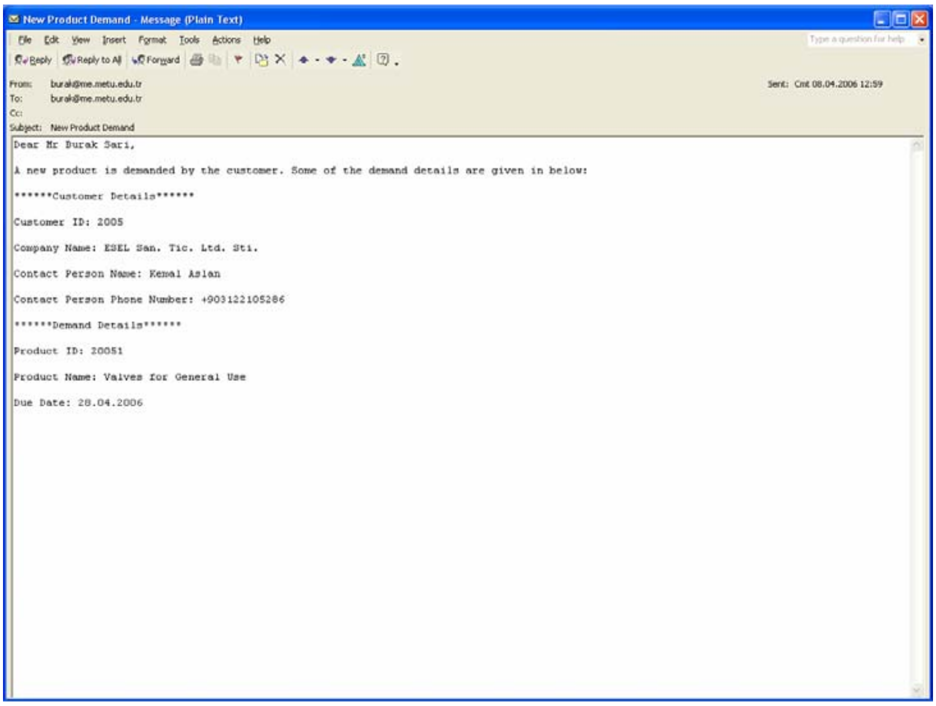


Figure 7.20 Notification e-mail about the product demand

7.1.2.4 Identify the customer requirements

Sample views from the identification of the customer requirements phase are shown on Figures 7.21 to 7.22. When a new product has been demanded, VE coordinator will examine the product demand and carry out a low level feasibility study. In this study, the feasibility of the product realization with the available resources will be determined. According to the result of this study, the coordinator will either accept the product demand and decompose the product into the relevant tasks or reject the product demand. If the product demand has been accepted, the administrator should need to detail the customer requirements to get a comprehensive basis for the product data. Designers will have to re-establish the required technical drawings using their own CAD-tools.

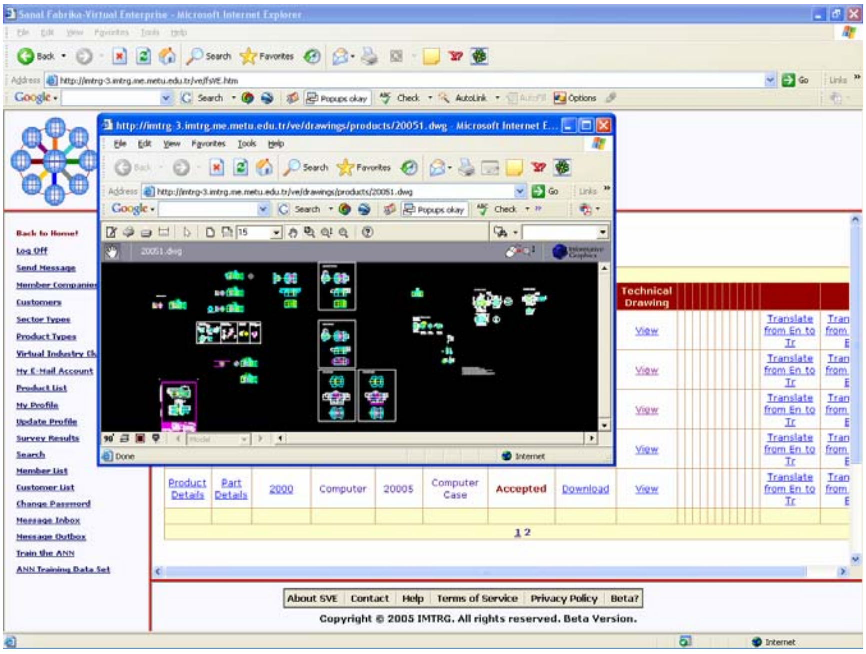


Figure 7.21 Product details

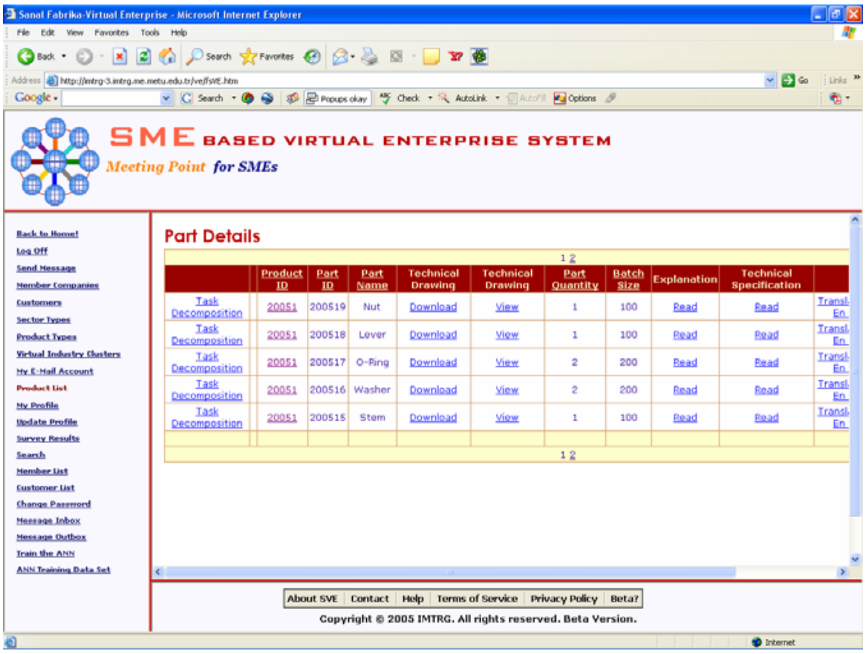


Figure 7.22 Part details

7.1.2.5 Task decomposition

Sample views from the task decomposition phase are shown on Figures 7.23 to 7.27.

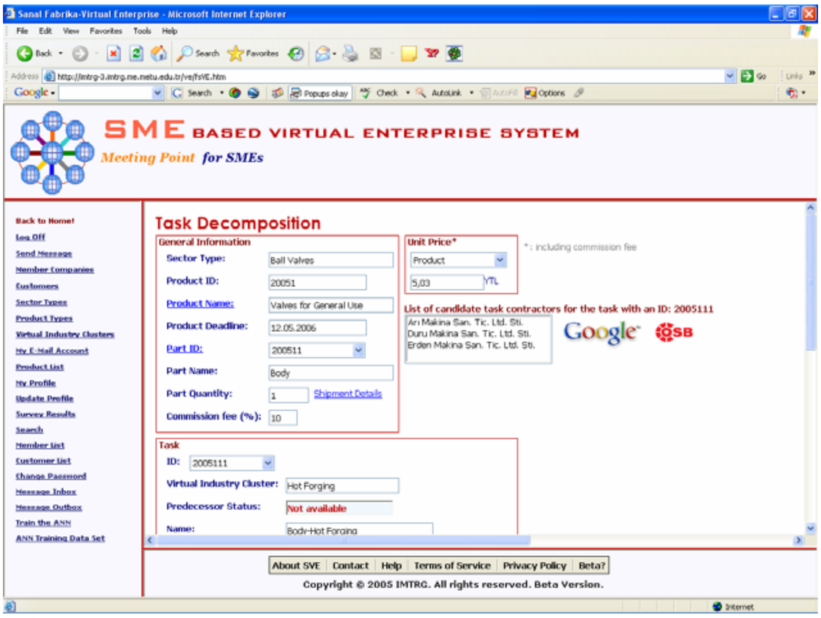


Figure 7.23 Task decomposition

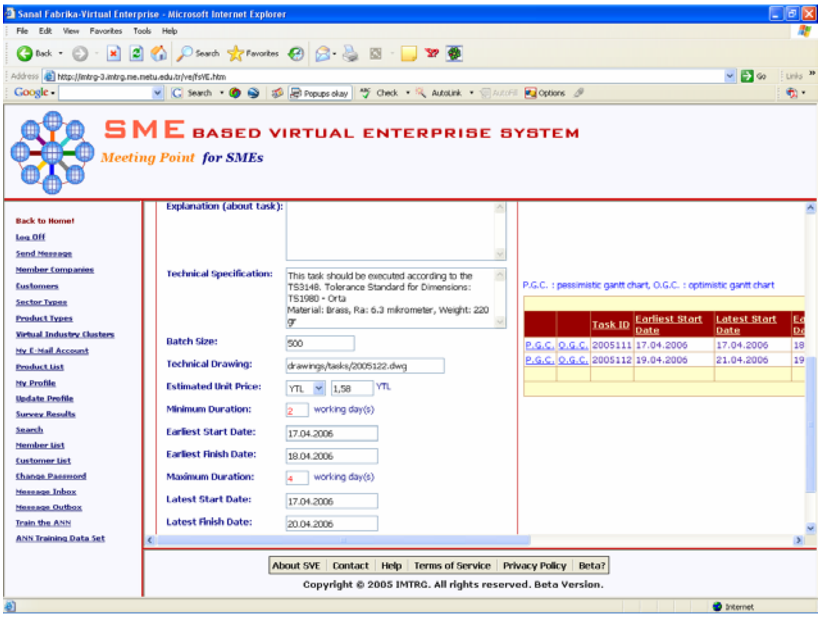


Figure 7.24 Task decomposition – continued

Once the product realization with the available resources in a requested time period has been found feasible, VE coordinator will decompose the product into tasks; identify the relevant tasks and different types of dependencies between pairs of tasks.

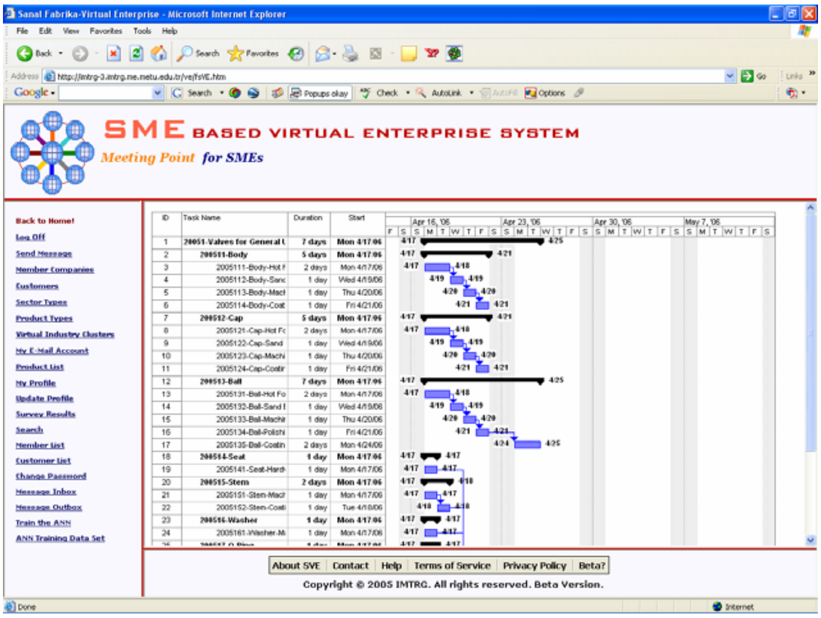


Figure 7.25 Optimistic gantt chart

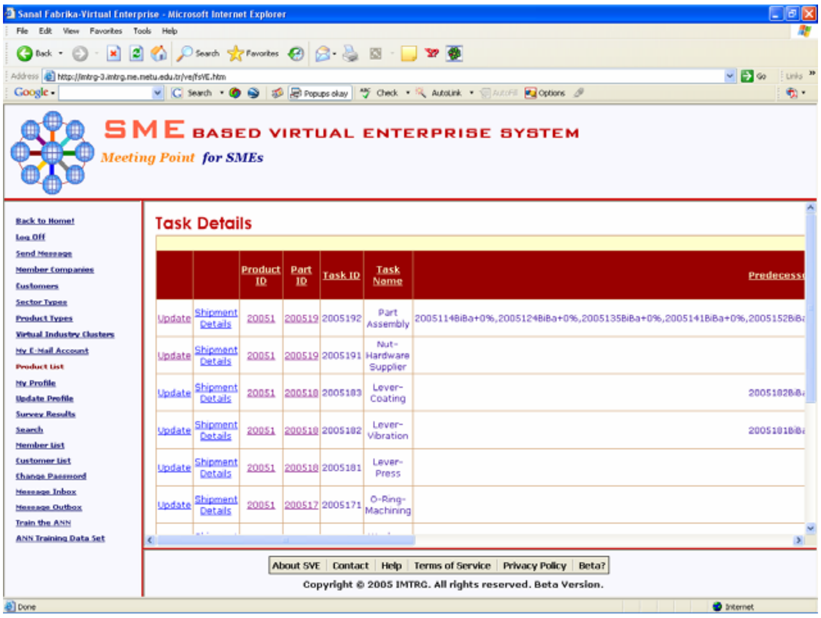


Figure 7.26 Task details

In the task decomposition stage, task scheduling will also be established based on the customer requirements. As an essential element in task decomposition, task scheduling aims to ensure that the tasks are completed on time. It includes definition and sequencing of tasks, task duration and cost estimation. Tasks due dates and different types of dependencies between successive tasks will also be determined.

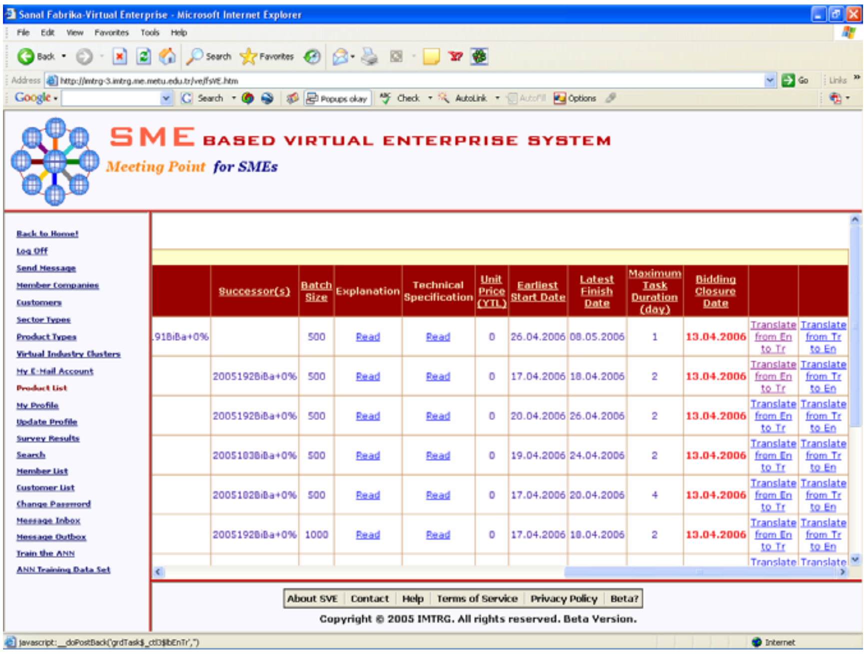


Figure 7.27 Task details - continued

7.1.2.6 Shipment details

Once the product has been decomposed into various tasks, VE coordinator should arrange the shipment details of these tasks.

In this phase, the coordinator should specify the target location of the shipment for each task and the batch size of the task to be shipped. The shipment time in terms of percent of task duration elapsed should also be provided in this phase.

Sample views from the shipment phase are shown on Figures 7.28 to 7.29.

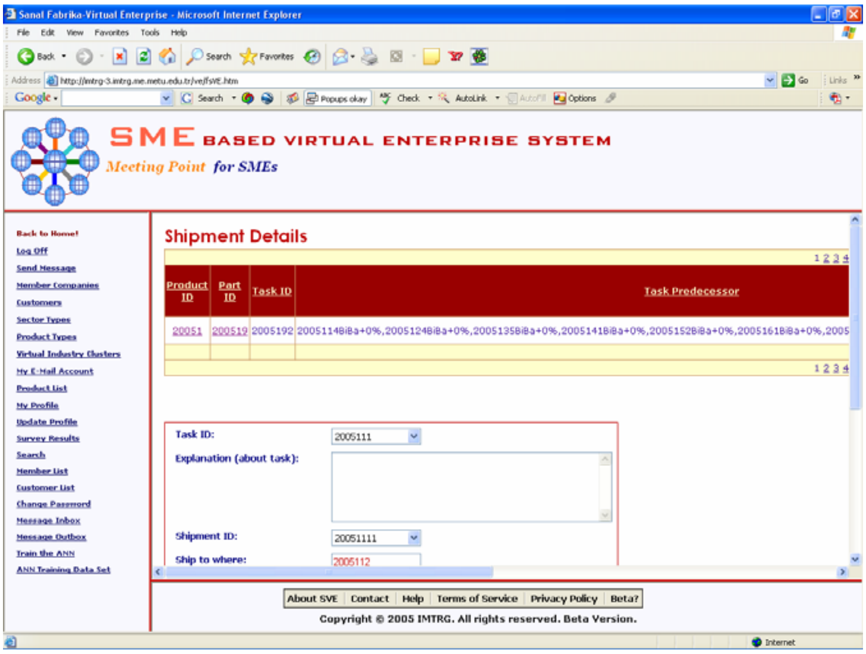


Figure 7.28 Shipment details

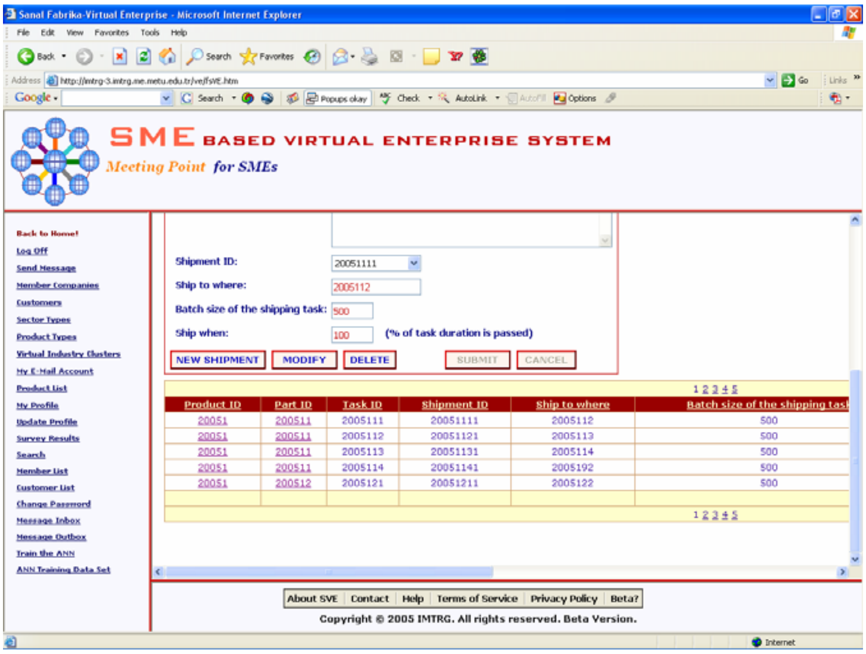


Figure 7.29 Shipment details – continued

7.1.2.7 Bid collection

Sample views from the bid collection phase are shown on Figures 7.30 to 7.35.

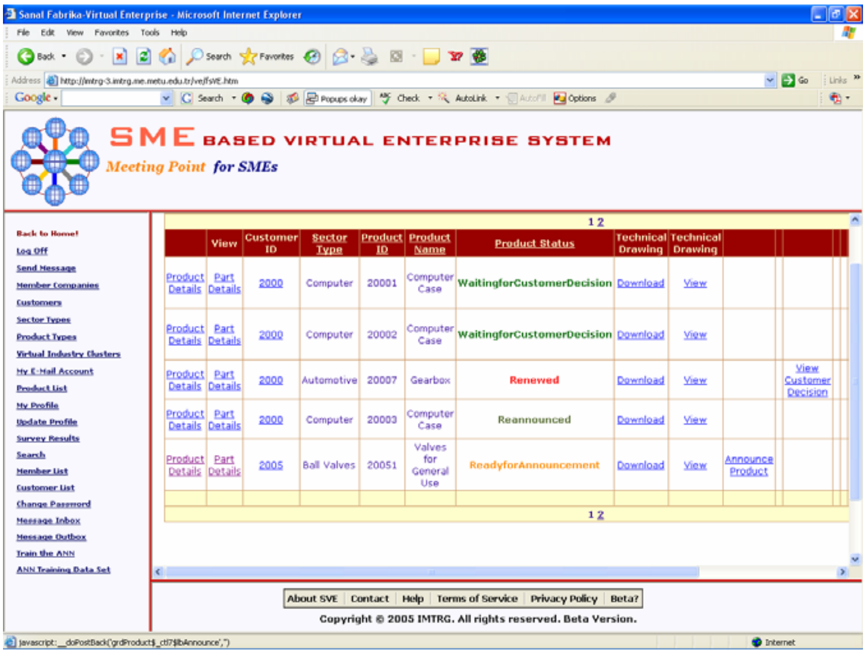


Figure 7.30 Product announcement

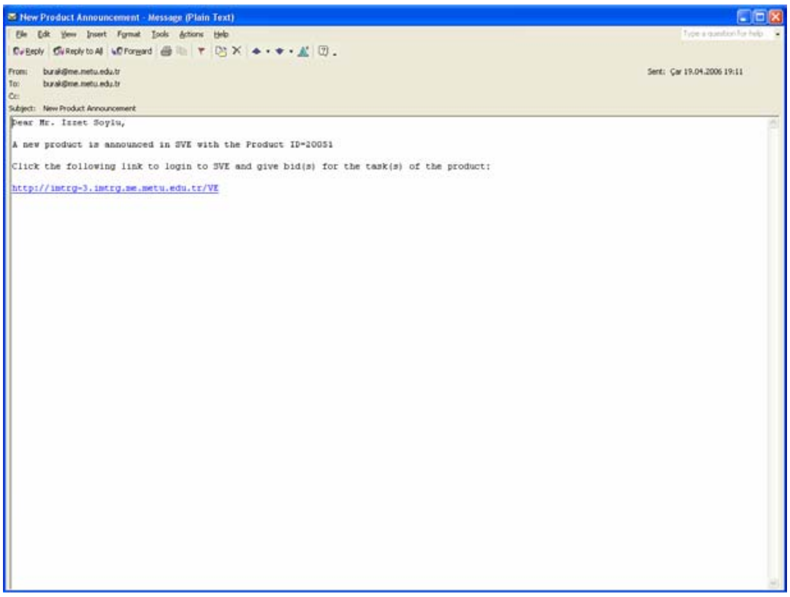


Figure 7.31 Notification e-mail about the new product announcement

Once the coordinator announces the tasks, bids will be collected from the members if their skill, resources, availability and cost bids are adequate to respond to the customer request.

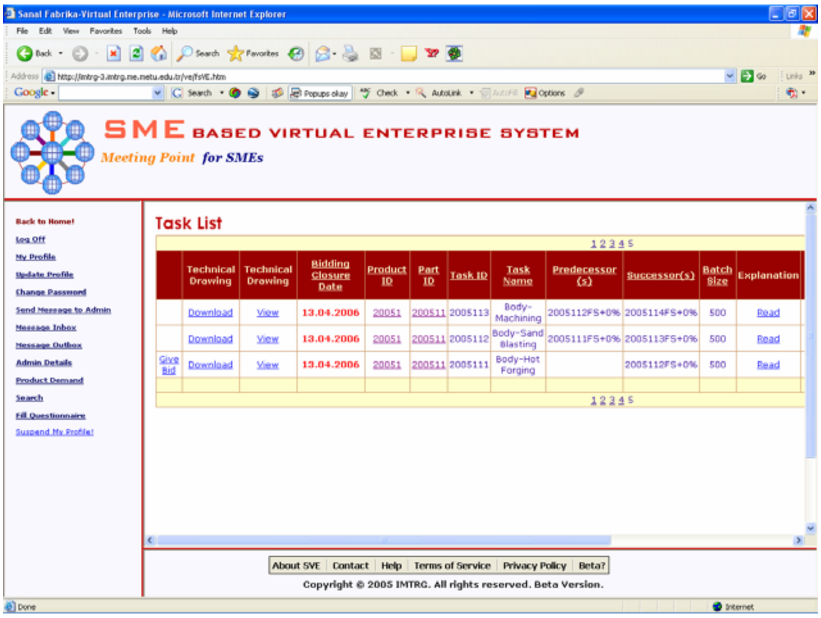


Figure 7.32 List of announced tasks

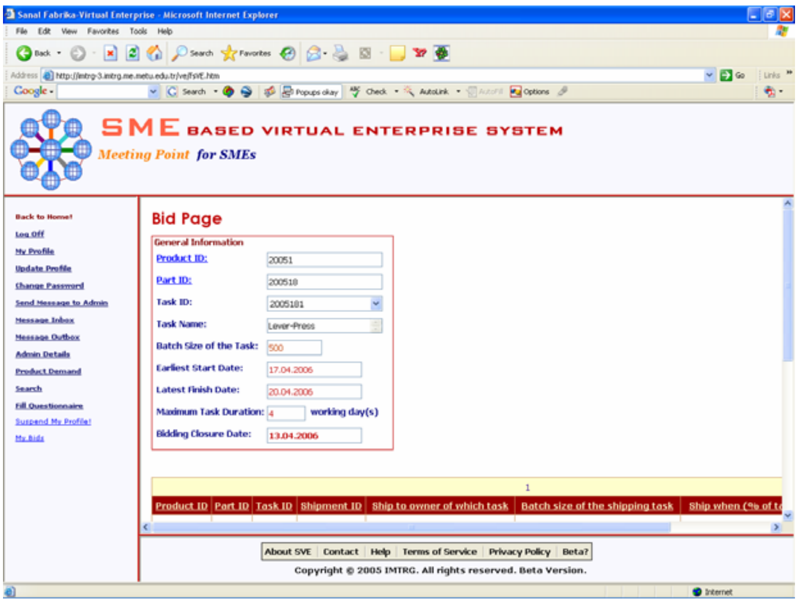


Figure 7.33 Bid page

Members can view only the tasks concerning them and submit a bid for the appropriate task. Note that the members should give their best bid to succeed in being a partner of the VE. If a specific task announcement can not be satisfied by the members' bids, then the coordinator should look for new members.

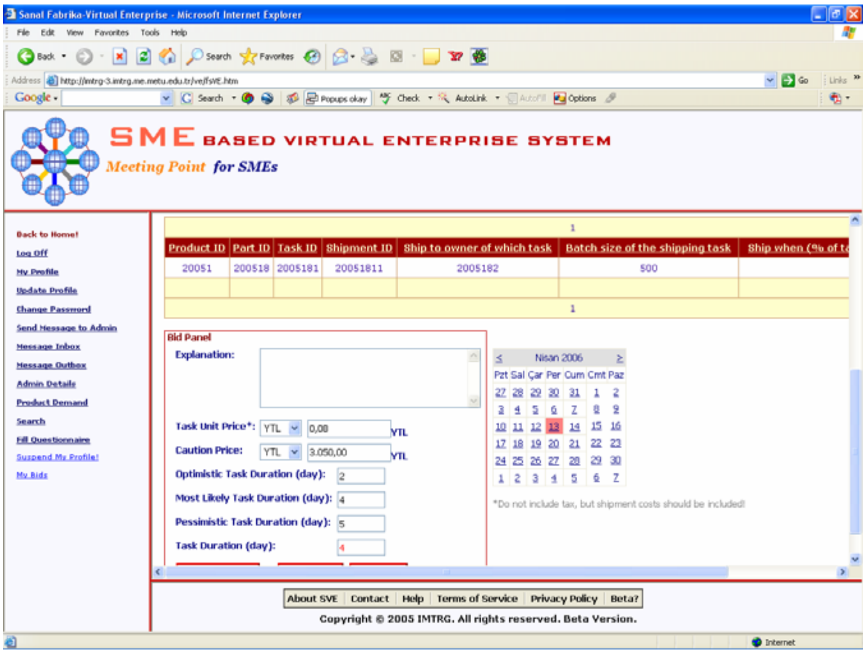


Figure 7.34 Bid page – continued

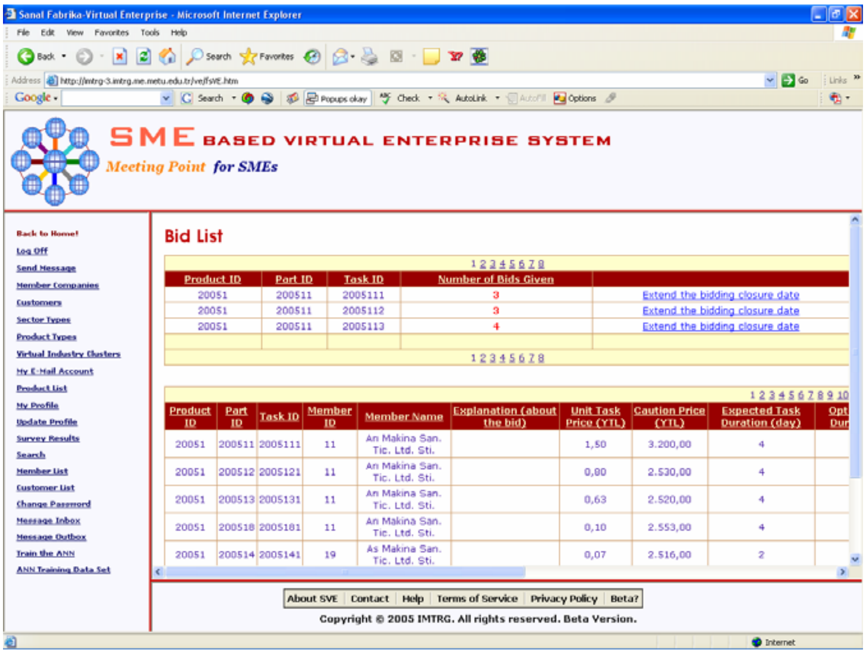


Figure 7.35 Bid list

7.1.2.8 Selection of the VE partners

Sample views from the selection phase are shown on Figures 7.36 to 7.41.

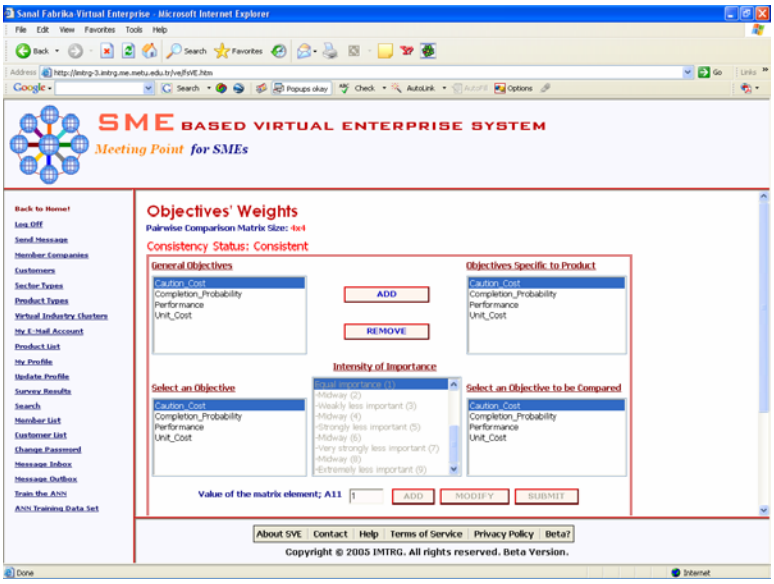


Figure 7.36 Objectives weights

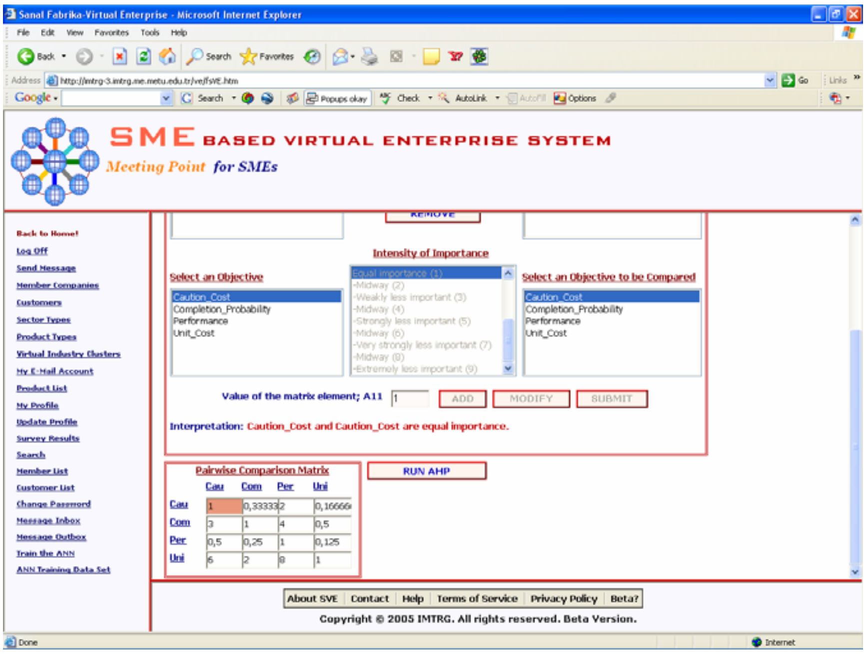


Figure 7.37 Objectives weights – continued

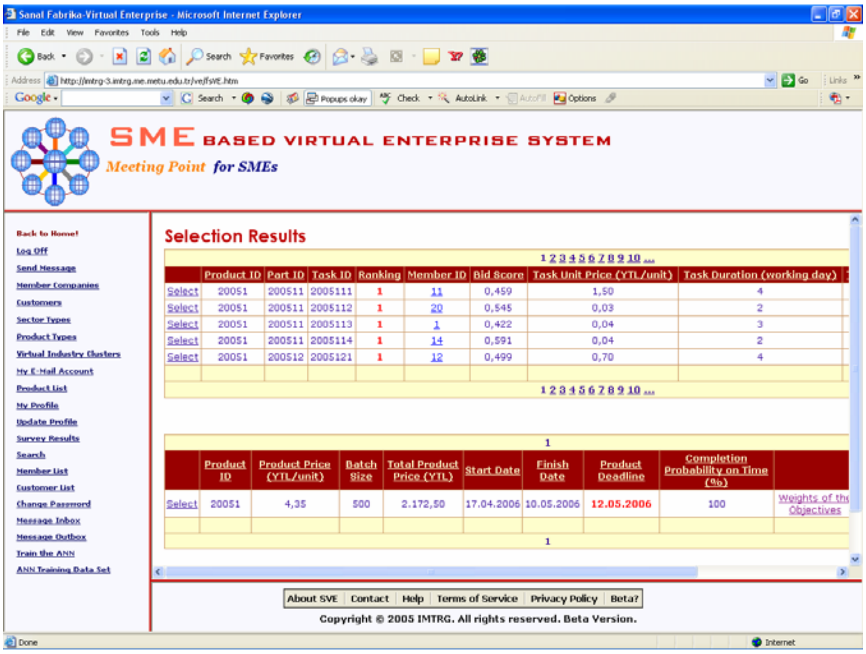


Figure 7.38 Selection results

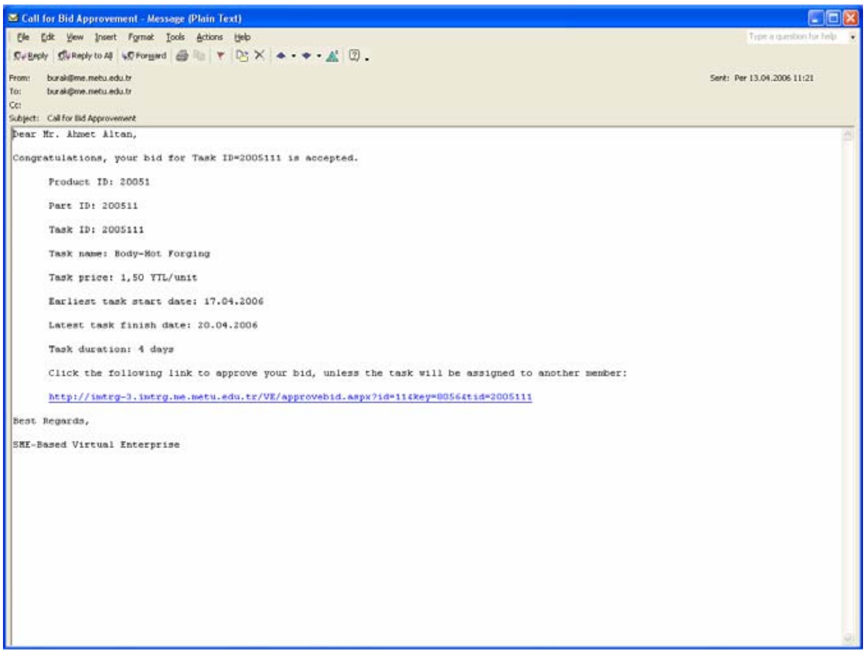


Figure 7.39 Call for bid approvement

Once the bids are collected, it's time to identify and select the partners that will participate in the virtual enterprise. Analytic Hierarchy Process (AHP) will be applied to rank order the multiple partner proposals. AHP assigns weights to each of the user selection criteria and then assigns weights the various proposal alternatives against each of these criteria.

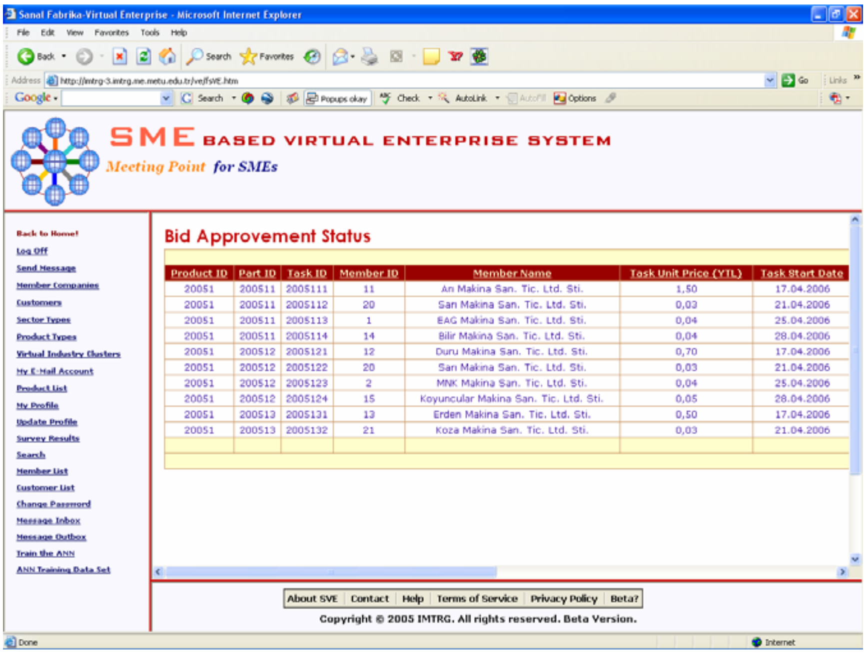


Figure 7.40 List of bid approvement

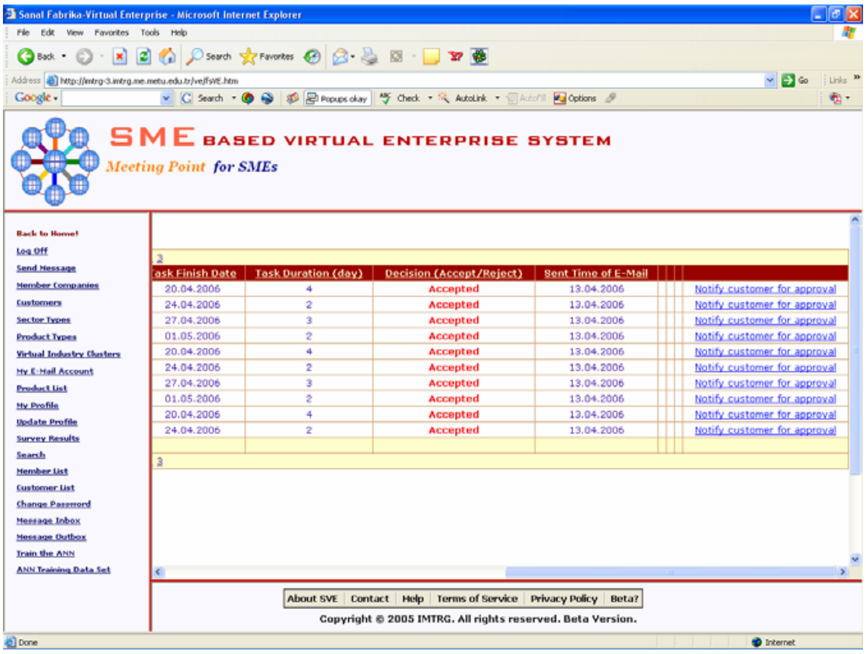


Figure 7.41 List of bid approvement – continued

In order to run AHP, first of all, the selection criteria should be determined. Secondly, a comparison should be done between the objectives to form a pairwise comparison matrix. This comparison includes the determination of the intensity of importance between the objectives.

Once the AHP algorithm is run, the result is a rank ordering of proposals that takes into account both the importance of each criterion to the user as well as the comparative evaluation of all alternatives against each of the selection criteria.

VE coordinator can either accept or reject the product demand according to the value of completion probability; being the probability of completing the product within the given time period. If it is found to be in an acceptable range, then the VE coordinator accepts the product demand and asks for an approval from the bid winners. Tender package will not be sent to the customer until all the bid winners are approved their own bids.

7.1.2.9 Negotiations with the customer

Tender package related with the product demand will be offered to the customer. The tender requires negotiations with the customer on the unit price or deadline of the product. Sample views from negotiation with the customer phase are shown on Figures 7.42 to 7.44.

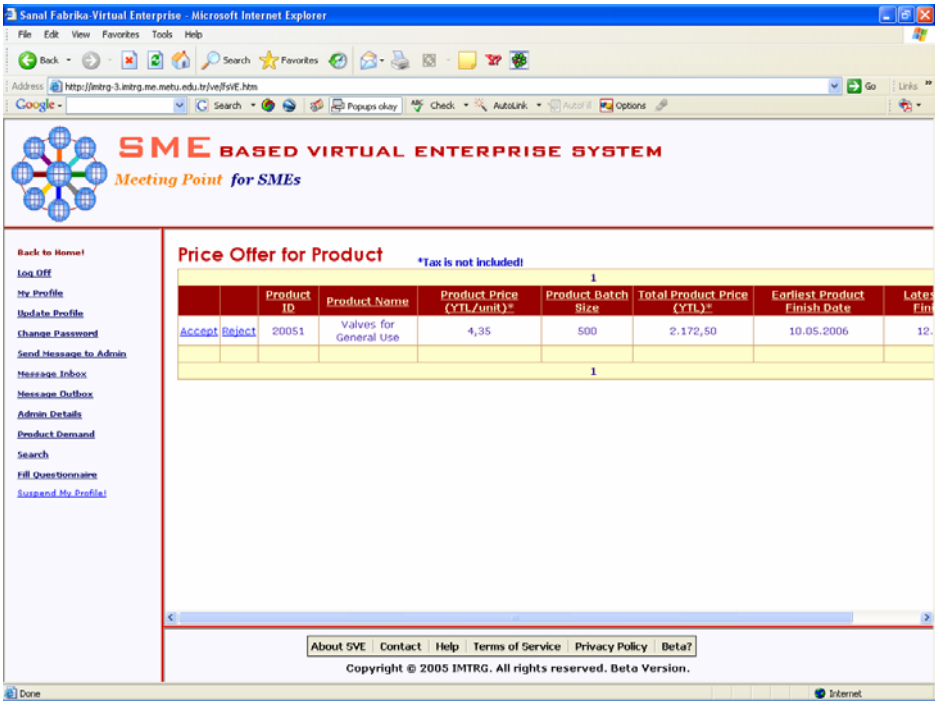


Figure 7.42 Price offer for product demand

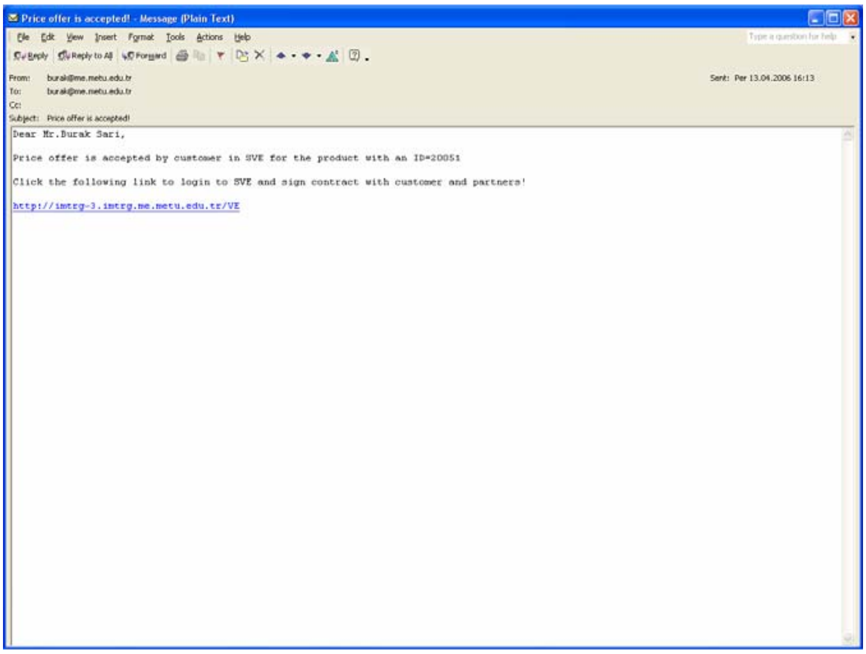


Figure 7.43 Notification e-mail for the acceptance of tender package

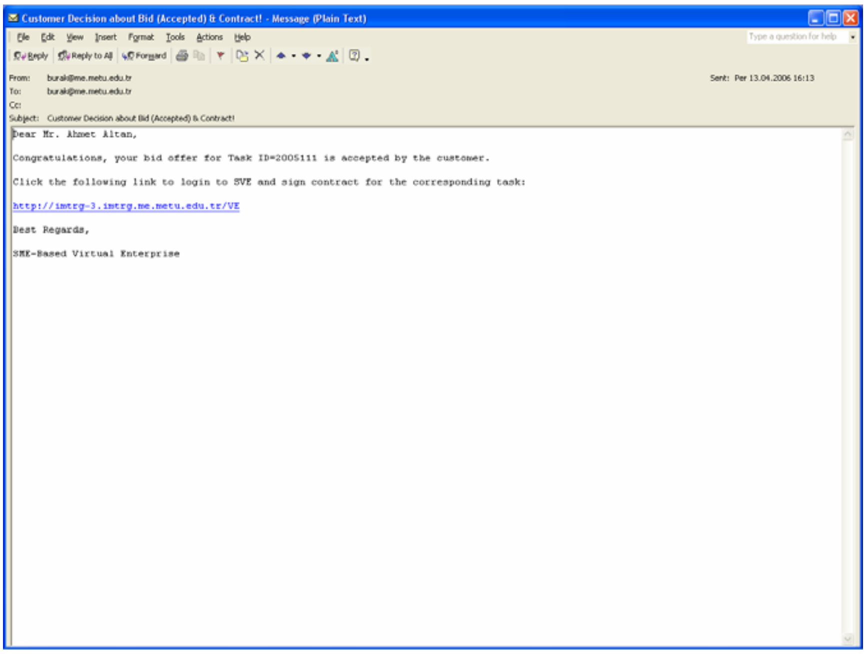


Figure 7.44 Notification e-mail for the acceptance bid offer

7.1.2.10 Establishment of the contracts with the partners & customer

Once the tender package has been accepted by the customer, at this stage contracts will be signed between the coordinator and customer and between partners and coordinator. VE acts as a mediator between the customer and the partners. Both sides will have a right to edit the contract templates. Sample views from the contract signing phase are given on Figures 7.45 to 7.50.

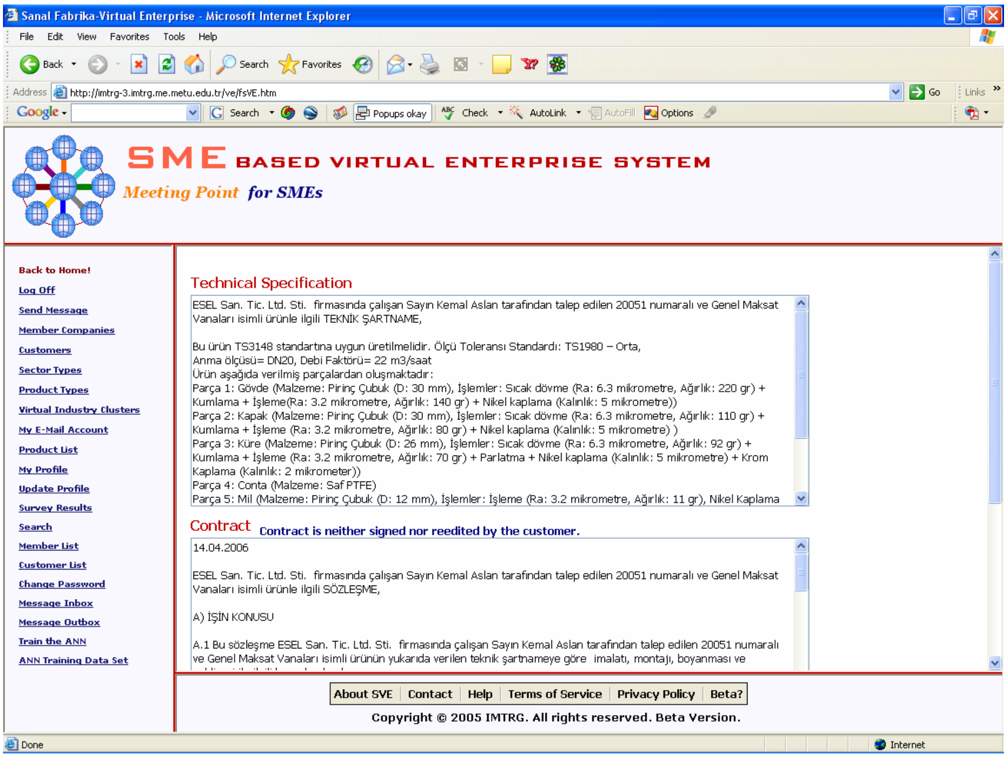


Figure 7.45 Product contract

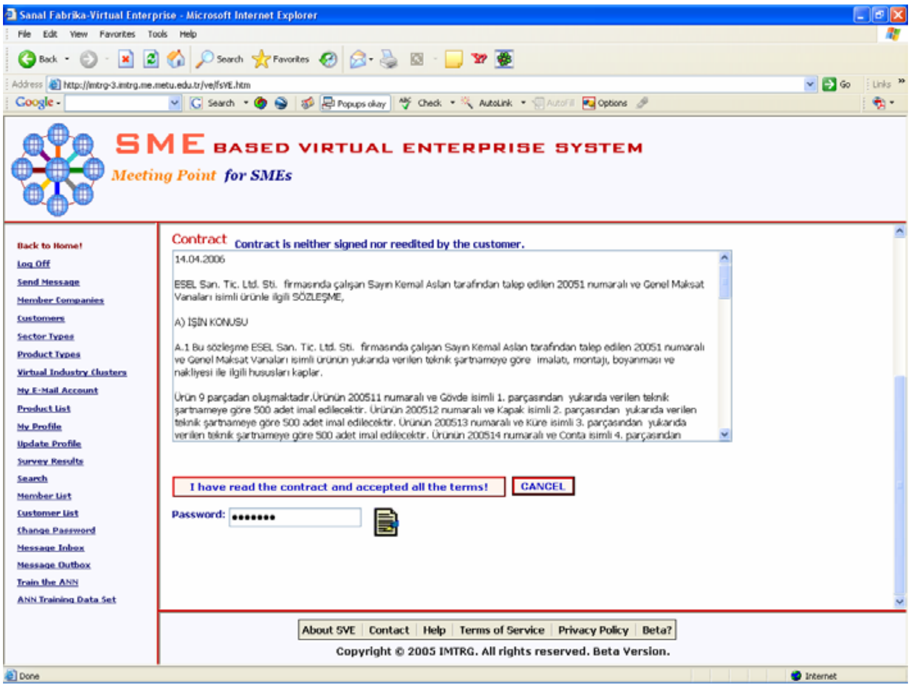


Figure 7.46 Product contract – continued

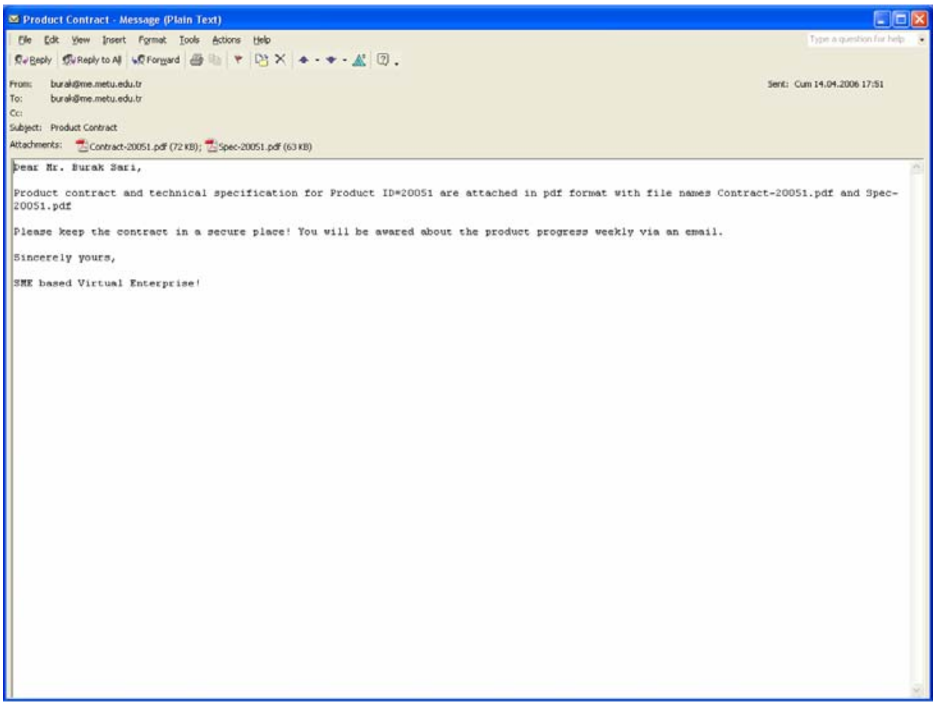


Figure 7.47 Attached product contract & specification

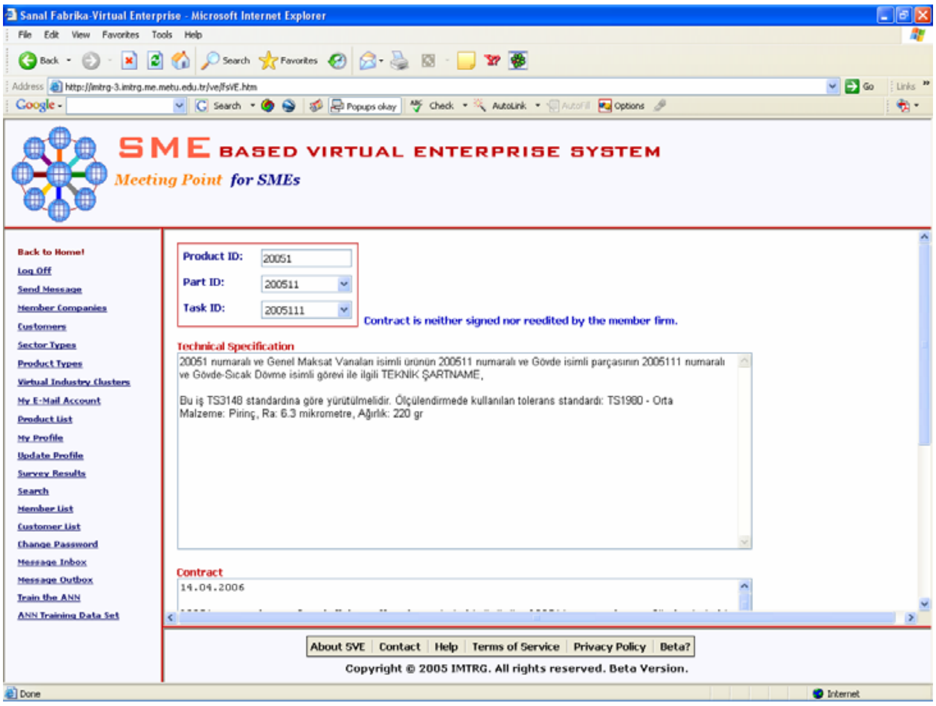


Figure 7.48 Task contract

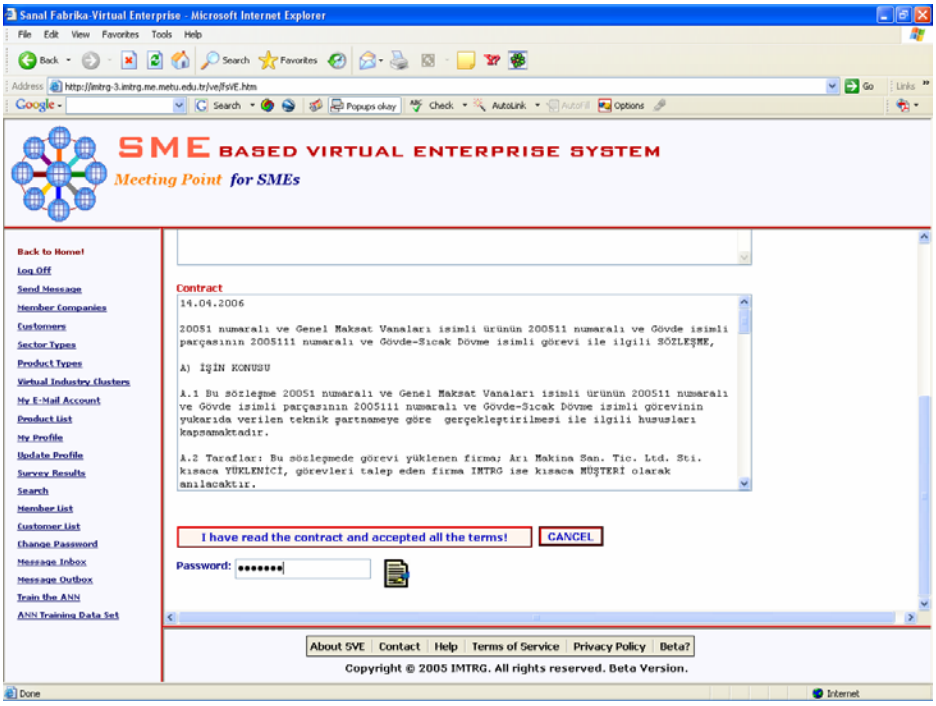


Figure 7.49 Task contract – continued

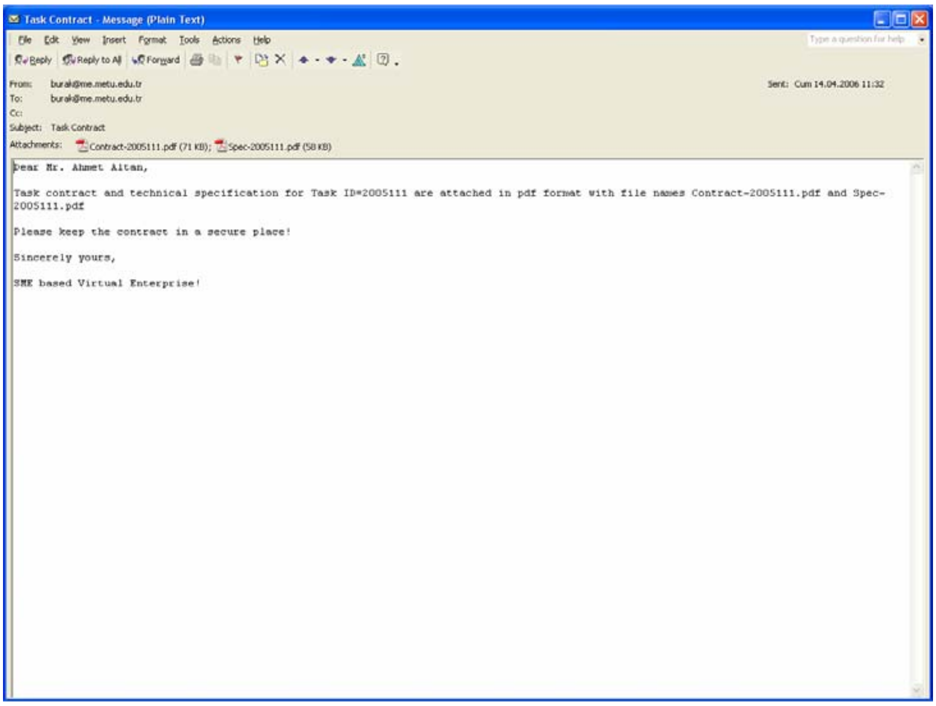


Figure 7.50 Attached task contract & specification

7.1.3 Operating the virtual enterprise

Once the VE has been set up and configured, it becomes functional and begins work to fulfill the customer needs.

7.1.3.1 Monitoring

Sample views related with the monitoring phase are given on Figures 7.51 to 7.54.

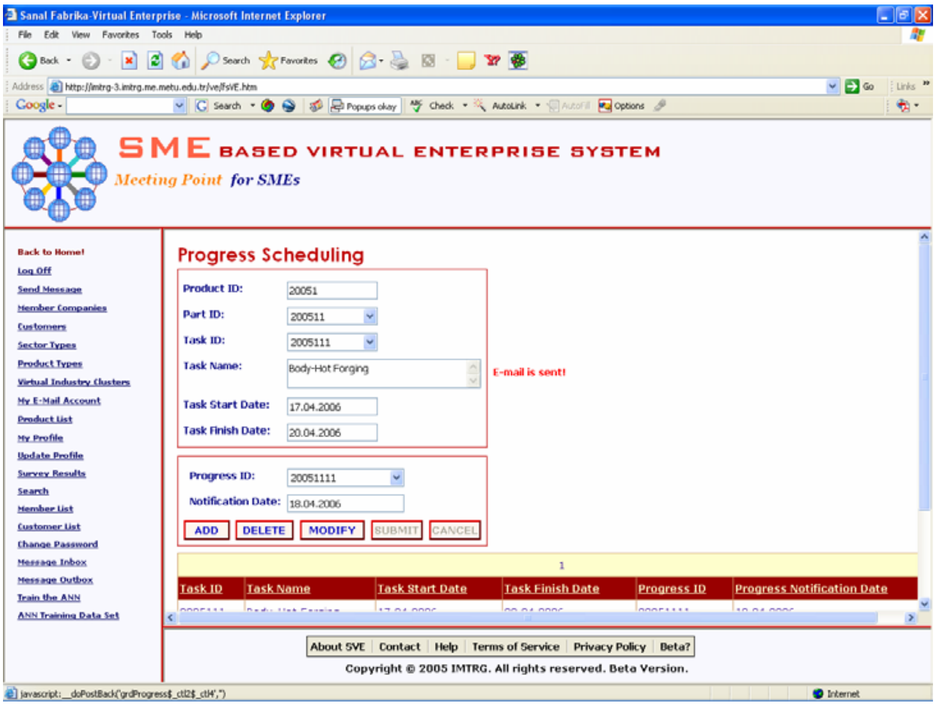


Figure 7.51 Progress scheduling

Monitoring together with progress reporting is an important project management activity. During the operation of a VE, different actors (e.g. coordinator, customer) may want to know the status of the VE or part hereof and the progress made so far. Coordinator and customer may, for example be highly interested in whether the project works out as expected in terms of physical work. General activities:

- The coordinator shall determine the reporting frequencies, e.g. every time a sub-task is completed and/or on a regular basis (e.g. once a week)

- Capture the progress from partner

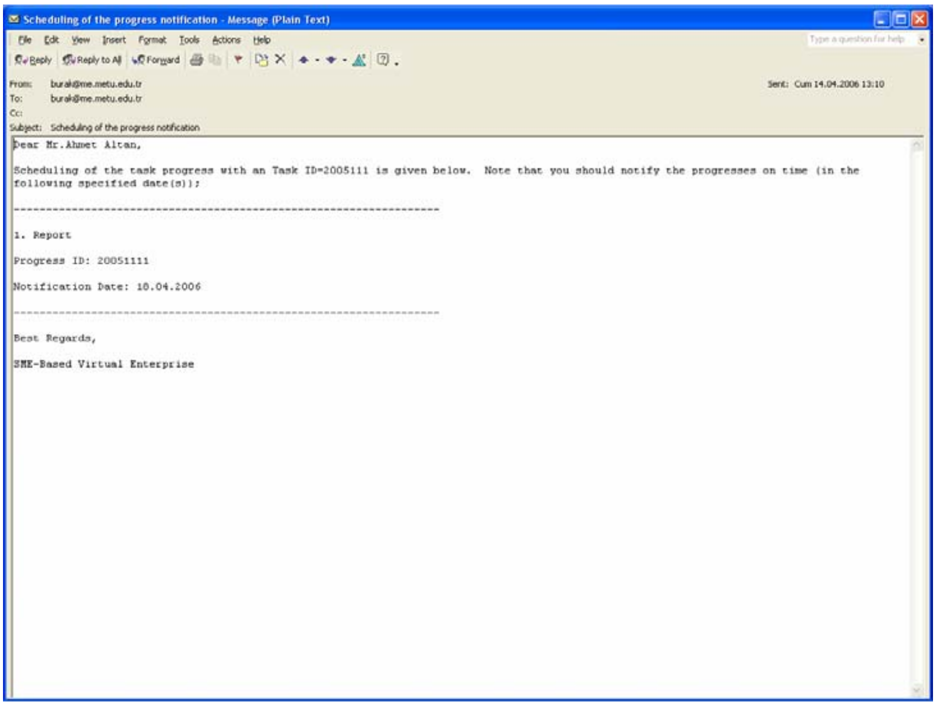


Figure 7.52 Notification e-mail for progress scheduling

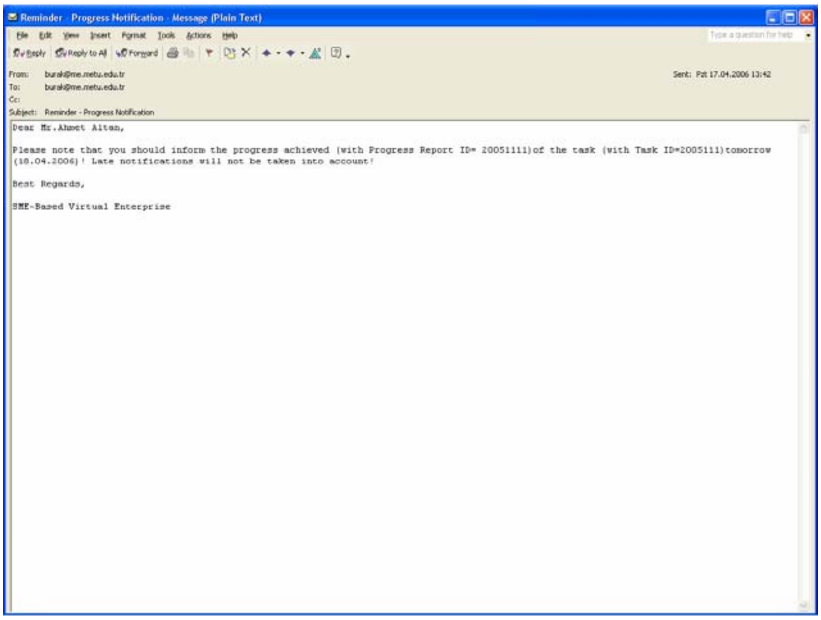


Figure 7.53 Reminder e-mail for progress reporting

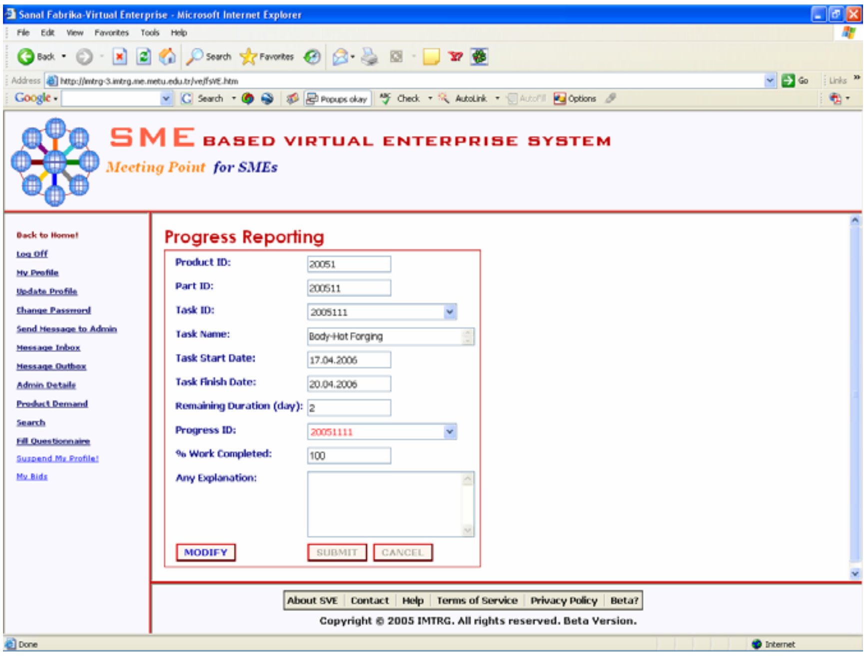


Figure 7.54 Progress reporting

7.1.4 Dissolution of VE

Sample views related with the dissolution phase are given on Figures 7.55 to 7.60.

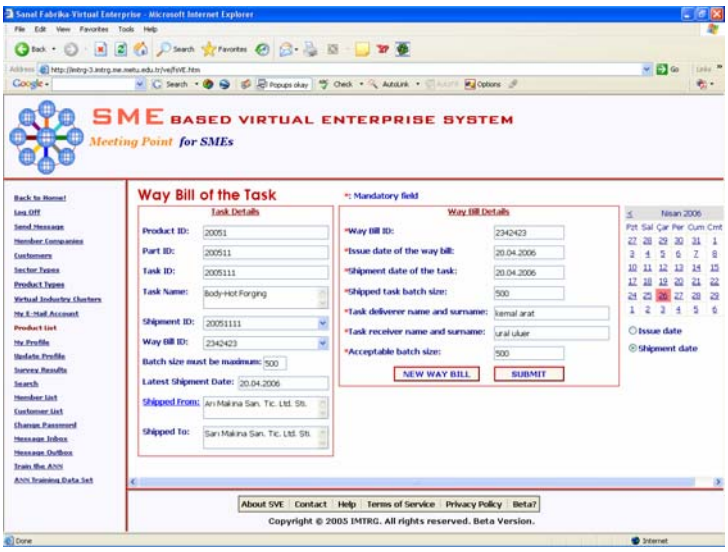


Figure 7.55 Fill of task way bill

Once the product has been completed, a final check should be done. Way bill & billing details of both the product and the tasks will be stored in the document repository of the system after the product has been handed over to the customer. Please note that partners' performance being achieved so far will be assessed for further partner selection processes during the entry of billing details of the tasks. When all paperwork has been completed and product payment is settled, the revenue will be divided according to the agreements between the VE and its partners.

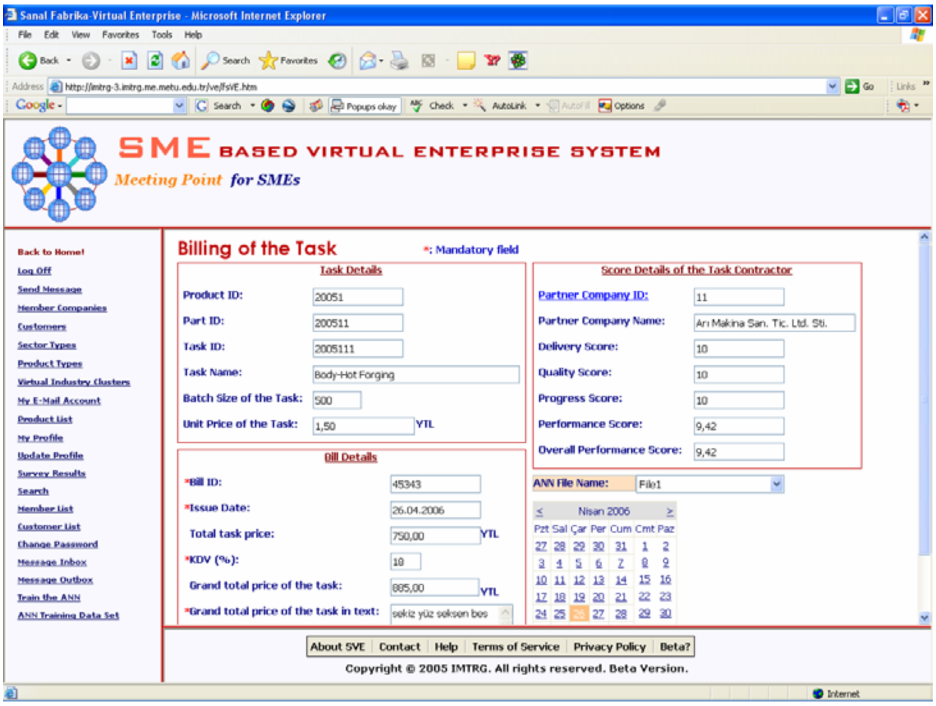


Figure 7.56 Fill of task bill & assessment of performance score

Before VE is closed down, the partners should give some time to assess system performance via questionnaire.

Once all the collected information is stored in documents repository for further use, the project is closed down and the partners go back to the virtual breeding environment and wait for new projects (VEs).

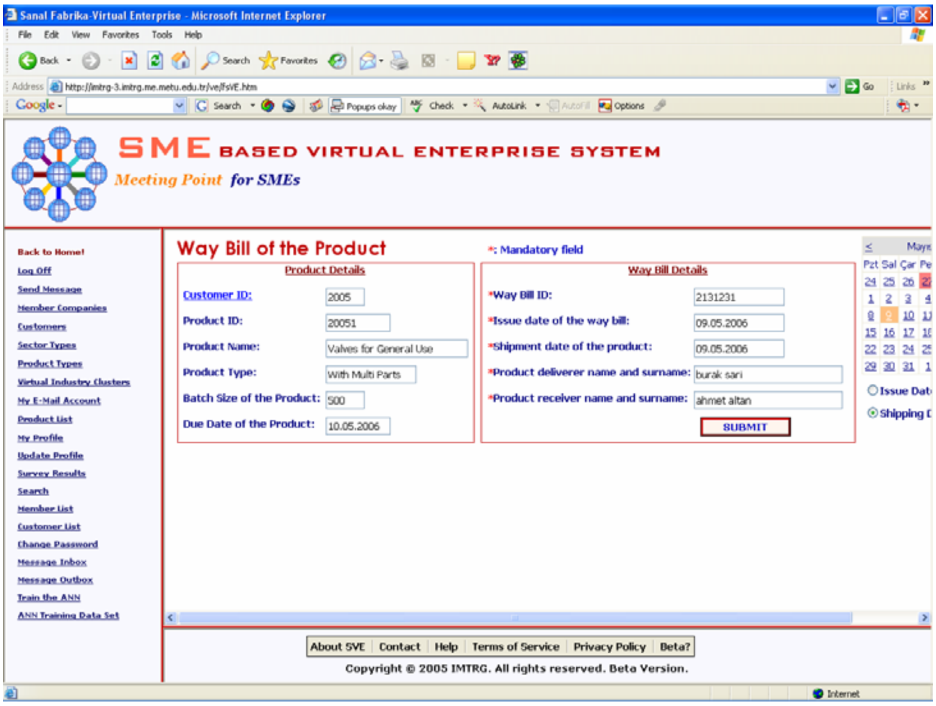


Figure 7.57 Fill of product way bill

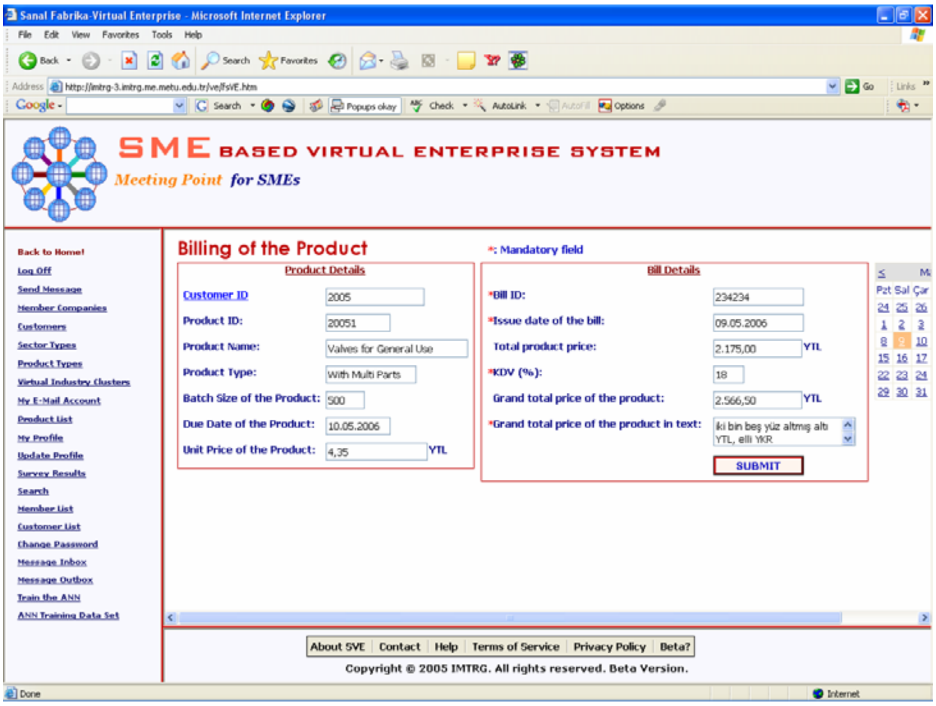


Figure 7.58 Fill of product bill

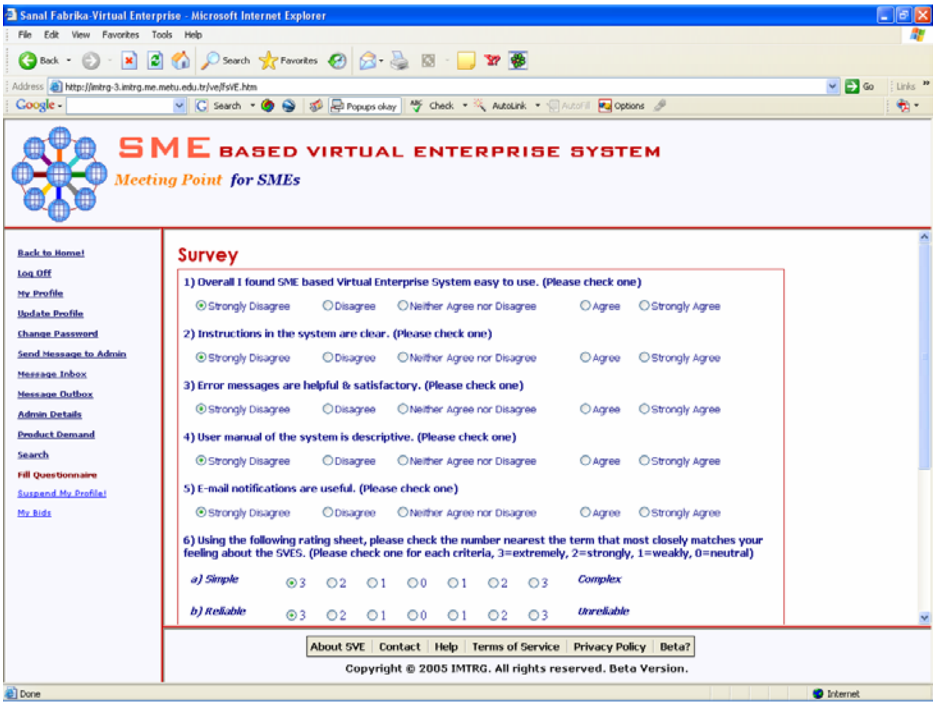


Figure 7.59 Survey

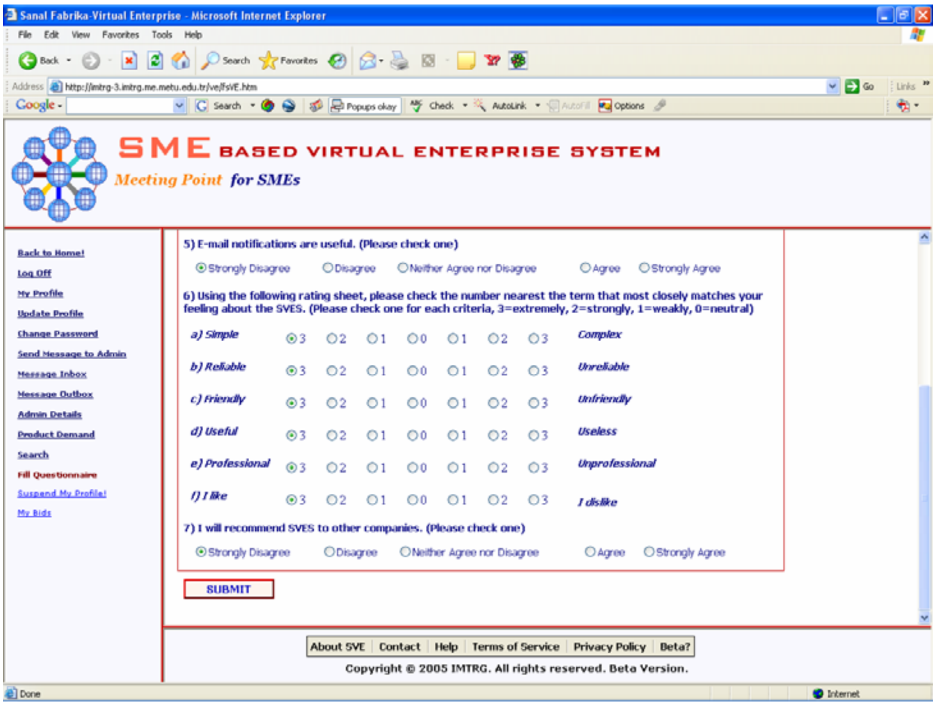


Figure 7.60 Survey - continued

7.1.5 Results

In this virtual case study, the production of ball valves for general use is realized with the attendance of various network members which would be responsible for the establishment of 23 diverse tasks in a given time period. In this scope, the product with multi-parts has been decomposed into 23 tasks and for each task an average of three bids were received from the members, thus 74 bids were collected in total. A total product price, 2175 YTL for a batch size of 500 units has been offered to the customer. Completion probability being the probability of completing the product by a given due date has been calculated as %100. Sample statistics related with evaluation of member bids are shown in Figure 7.61.

1 2 3

Product ID	Part ID	Task ID	Ranking	Member ID	Bid Score	Task Unit Price (YTL/unit)	Task Duration (working day)	Task Start Date	Task Finish D
20051	200511	2005111	1	11	0,459	1,50	4	17.04.2006	20.04.2006
20051	200511	2005112	1	20	0,545	0,03	2	21.04.2006	24.04.2006
20051	200511	2005113	1	1	0,422	0,04	3	25.04.2006	27.04.2006
20051	200511	2005114	1	14	0,591	0,04	2	28.04.2006	01.05.2006
20051	200512	2005121	1	12	0,499	0,70	4	17.04.2006	20.04.2006
20051	200512	2005122	1	20	0,545	0,03	2	21.04.2006	24.04.2006
20051	200512	2005123	1	2	0,499	0,04	3	25.04.2006	27.04.2006
20051	200512	2005124	1	15	0,545	0,05	2	28.04.2006	01.05.2006
20051	200513	2005131	1	13	0,538	0,50	4	17.04.2006	20.04.2006
20051	200513	2005132	1	21	0,548	0,03	2	21.04.2006	24.04.2006
20051	200513	2005133	1	3	0,514	0,07	3	25.04.2006	27.04.2006
20051	200513	2005134	1	22	0,36	0,04	2	28.04.2006	01.05.2006
20051	200513	2005135	1	16	0,452	0,10	4	02.05.2006	05.05.2006
20051	200514	2005141	1	18	0,512	0,05	2	17.04.2006	18.04.2006
20051	200515	2005151	1	1	0,514	0,16	3	17.04.2006	19.04.2006
20051	200515	2005152	1	14	0,367	0,07	2	20.04.2006	21.04.2006
20051	200516	2005161	1	5	0,267	0,03	2	17.04.2006	18.04.2006
20051	200517	2005171	1	6	0,336	0,05	2	17.04.2006	18.04.2006
20051	200518	2005181	1	17	0,508	0,08	4	17.04.2006	20.04.2006
20051	200518	2005182	1	19	0,545	0,03	2	21.04.2006	24.04.2006
20051	200518	2005183	1	16	0,515	0,07	2	25.04.2006	26.04.2006
20051	200519	2005191	1	18	0,548	0,03	2	17.04.2006	18.04.2006
20051	200519	2005192	1	4	0,408	0,08	3	08.05.2006	10.05.2006
20051	200511	2005111	2	12	0,274	1,58			
20051	200511	2005112	2	22	0,232	0,04			

1 2 3

Figure 7.61 Bid statistics of the case study

7.2 Validation Platform

The findings in the research have been validated throughout the research by the various activities as presented in Table 7.3. Applying a combination of the validations activities have resulted in a triangulation of the results in a way that the combination of the various validation activities to strengthen the findings.

Table 7.3 Validation platform

Activity	Description
Reviews	As a means for the faculty to ensure the quality of Ph.D. research, a group of assessors are assigned to follow the progress and results of the research. As a consequence 5 reviews have been carried out during the research. Another type of reviews of the results has been conducted as a part of the traditional acceptance process of papers in relation to conferences proceedings, and journal papers.
Conferences	Another means for validation of the results has been through participation in international conferences. This includes both before mentioned review process in relation to have the papers accepted for proceedings as well as the discussion with peer conference delegates in general and in relation to the presentation in particular.
Presentations	The findings have been presented in various settings throughout the duration of the project. This includes above mentioned progress meetings and conferences.
Journal Publications	Several journal articles have been prepared to publish the findings. Some of them are still in review. Similar to conferences, this also includes review process in relation to have the papers accepted.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

Enterprises, not least in one-of-a-kind industries, face major challenges as regarding full exploitation of existing and future ICT potentials. Major keywords for the challenge are globalization, outsourcing and customization, ref. section 3.1. The vision is, with an enterprise network as a breeding environment, to rapidly set up customer-focused virtual enterprises with partners each mastering complementary competencies, which, in joint correlation, can create products fulfilling customer needs. This demands a new global understanding enabling a safe cross-company communication and negotiation in the preparation for and the order-triggered set up of virtual enterprises.

Taking outset in a clarification of the background question why virtual enterprises are relevant this research has examined four research questions (RQs) in relation to virtual enterprise: RQ1: What is a virtual enterprise, RQ2: What type of knowledge is needed to realize VEs? (and how could this knowledge be structured?), RQ3: How to prepare, set up and operate VEs? (applying the identified types of knowledge), and RQ4: How to test and validate VE methodology? The RQs are shown in Figure 8.1 along with the main results of this Ph.D. related to the four questions.

One of the main challenges in relation to enterprise networks and virtual enterprise is to be able to set up virtual enterprise fast and efficient. To put it simply: it is to be able to create a virtual enterprise composed of the best available competencies, but if the setup cannot be done within a competitive timeframe the raison d'être of the virtual enterprise vanishes. Thus, there is a need for collecting and structuring the knowledge that can facilitate the setup and operation of enterprise networks and their accompanying virtual enterprises. As a response to this challenge the ICT reference architecture for virtual enterprise systems and VE methodology (VEM) have been developed as a part of this Ph.D. research. ICT reference architecture runs using the n-tier client/server technology and structured methodology that can be used when designing and implementing such systems for industrial implementations or academic research purposes. VEM guides (virtual) enterprise engineers in preparing, setting up and operating virtual enterprise.

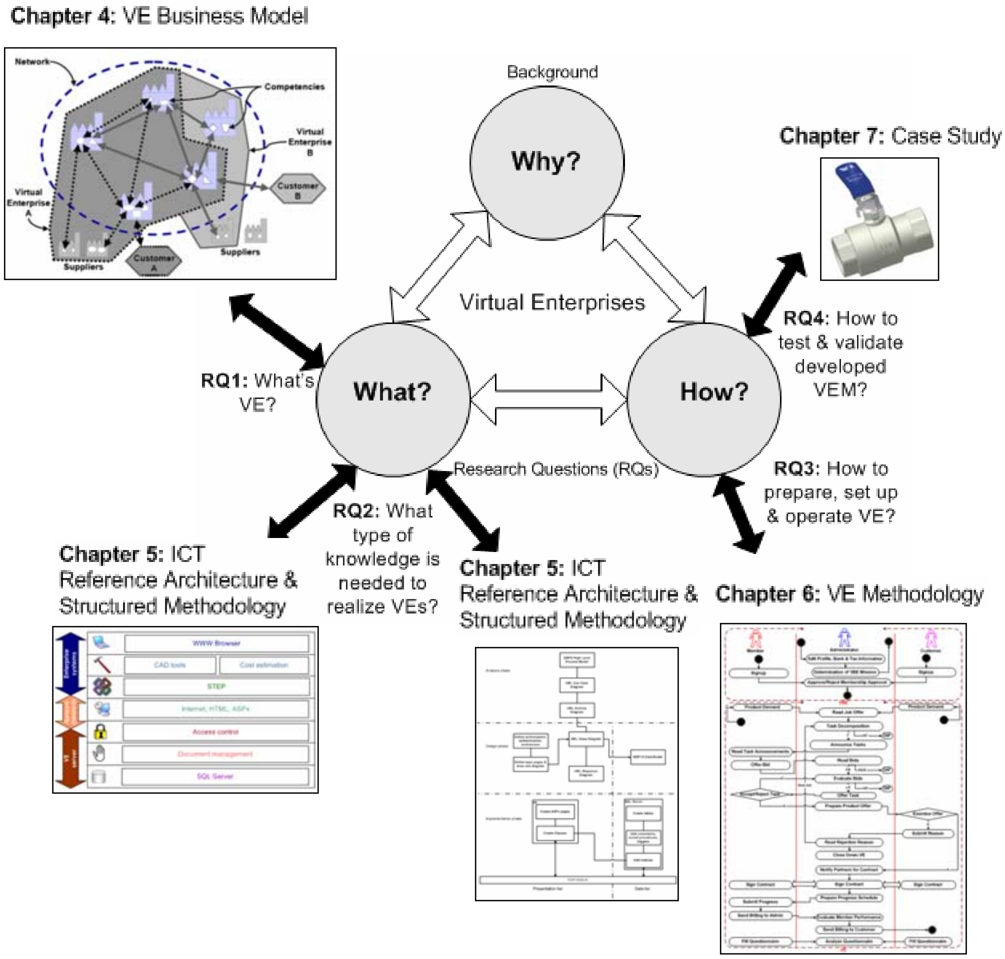


Figure 8.1 Research questions and main results

In section 8.1 summarization and discussion of the findings of this research in relations to the research questions are provided. A summary of the contributions of the research is presented in section 8.2. In section 8.3 the research limitations are addressed. Finally future work is addressed in section 8.4.

8.1 Discussion and Summary of Results

8.1.1 Research question 1 – what is a VE?

When the virtual enterprise concept emerged in the early 1990s it initially received attention as being perceived as a new organizational form in the virtual marketplace where new and unknown partners in a plug and play manner could cooperate by means of the new and advanced information and communications technologies. These expectations have not been fulfilled.

Issues such as trust as well as the risk and cost related to dealing with unknown partners have been pointed out as some of the obstacles for fulfilling the virtual enterprise in its more 'purest' theoretical form.

This research pursues to examine if the virtual enterprise business model is necessary at all or if it can be applicable in a dynamic business environment. The first question in this research is:

> *RQ1: What is a virtual enterprise? Is there a suitable business model for enterprises operating in a dynamic inter-enterprise environment?*

The virtual enterprise business concept as described in section 3.2.7 characterizes the virtual enterprise as a temporary entity established by autonomic enterprises possessing complementary competencies enabled by information and communications technology.

In this dissertation, virtual enterprise is defined in the following way:

> *"A Virtual Enterprise is a customer solution delivery system created by a temporary and re-configurable ICT enabled aggregation of core competencies."* (Sari et al, 2004)

The perception of a virtual enterprise as a new entity established in a fast and efficient manner among partners with little or no joint cooperation experiences has been pointed out as a problem. When presented, the industrial partners have accentuated the issue of trust as an obstacle for cooperation with partners among whom no cooperation experience exists. Thus, the sole virtual enterprise entity as such cannot be perceived as a inclusive feasible business model in a dynamic inter-enterprise environment where the partnership are expected to be changed from project to project.

Enterprise network works as a breeding environment for setting up virtual enterprises. In the enterprise network long-term partnerships can be nurtured and furthermore a degree of preparedness can be established enabling a fast and efficient setup of virtual enterprises based upon particular customer needs.

Thus, the VE business model as presented in chapter 4 and depicted in Figure 4-1 represents an applicable business model for enterprises engaged in a dynamic inter-enterprise environment in line with the virtual enterprise. The model bridges the dynamic and agile aspects of the virtual enterprise with the more stable and long term properties of the enterprise network, which again have familiar characteristics with the extended enterprise.

8.1.2 Research question 2 – what framework is needed?

A fast and efficient setup and operation of virtual enterprises can be facilitated through preparedness in the enterprise network. In order to establish such a preparedness one need an overview of the type of knowledge relevant in relation to virtual enterprise set up, preparation, and operation. Accordingly, the second question in this research is:

> *RQ2: What framework is needed to structure the body of knowledge related to preparing, setting up and operating virtual enterprises?*

In pursuing to answer the question a framework and ICT reference architecture have been developed as a part of this research. The reference architecture developed is the ICT Reference Architecture presented in chapter 5. The framework is the Structured Methodology which is also presented in chapter 5.

When proposing the ICT reference architecture, three levels with seven layers are proposed to portray in a diagrammatic fashion how different enterprises may exchange and use information between their respective organizations' specific proprietary systems and a central server. The lowest level (server level) constitutes the "central" server domain. It contains three distinct layers. Storage layer is the one which describes databases, backups, etc. In short, this is the server repository. Service layer describes the services which are made available through the server. Access layer is the one which describes the different forms and mechanisms available for access control to the server. The middle level, network constitutes the path and mechanism through which organization specific applications/environments of an end user would interact with the server. It consists of one distinct layer, communication layer, which describes the various protocols, methods, and formats through which an end-user or application could potentially communicate with the server. The upper level, enterprise systems level, constitutes the working environment and systems of the end-user through his/her organization specific applications and platforms. It consists of three layers. Application layer describes the different organization specific applications. Presentation layer describes the different interfaces and mechanisms through which a user can physically interact with an application. Interoperability layer describes the mechanisms and standards through which application specific information is restructured or mapped to comply with mutually agreed upon standards for data and information exchange.

The developed structured methodology has three phases. The first phase is the analysis phase where the system that will be re-engineered or the system that will be designed is analyzed and knowledge is extracted to be used for the design phase. In this phase, functional models, activity models and use cases of the system are defined. In the design phase, for all tiers of the system a detailed design activity is carried on. The details and specifics of this design activity are highly dependent on the features of the client/server technology of Microsoft .NET. This stage goes on the two tiers of the n-tier client/server model of Microsoft .NET and produces a data model that is ready for implementation in a database, object model that is ready for implementation using programming languages like VB .NET, sequence models to describe the different interactions between the classes, and a site model, which can be the basis of a web based site of the system for user interface. In the last stage, the implementation and testing activities are carried out again in two tiers using the tools and development environments of Microsoft .NET solution.

8.1.3 Research question 3 – how to prepare, set up & operate VEs?

To support the realization of VEs there is a need for a comprehensive methodology for VE engineering and management. This has been the focus of the third research question:

RQ3: How to prepare, set up, and operate virtual enterprises?

A partial and summarized version of the virtual enterprise methodology (VEM) has been presented in chapter 6 as a candidate for a methodology fulfilling RQ3.

VEM outlines relevant considerations for enterprises pursuing to prepare, set up, and operate virtual enterprises in accordance with the VE business model presented in chapter 4. VEM applies the comprehensive structured methodology presented in section 5.2 as its basis underlying structure. Structured methodology provides a generic structure, which permits a systematic approach to the complex and multi-dimensional tasks involved in creating virtual enterprises.

The methodology focuses on setting up and managing enterprise networks and virtual enterprises including the creation of customer delivery systems as part of the VEs operation. The methodology supports enterprises that are faced with the challenge of operating in accordance with the VE concept presented in relation to

RQ1. The VEM helps these enterprises to 'ask the right questions at the right time', and as such facilitates the planning and preparation of VEs.

VEM addresses activities of relevance when setting up and managing enterprise networks and VEs. The methodology integrates existing methods and procedures into a VE context, i.e. most of the management activities are based on well-known methods and procedures – what is new is that they are put into the VE context. Thus, the VEM works as an important means to widespread realization of more agile virtual enterprise type of organizations.

A modular and layered approach was used to define the ICT architecture as a part of the VEM based upon a consistent set of architectural best practices derived from the layered ICT reference architecture presented in section 5.1. This basis was drawn from the n-tier client/server technology of Microsoft .NET framework.

8.1.4 Research question 4 – how to test & validate the developed VEM?

One of the aims of this research is to carry out certain test cases in order to test the proposed VE methodology and validate the system by the establishment of various activities. Accordingly, the last question in this research is:

RQ4: How to test & validate the VEM?

To demonstrate the application of the developed VE methodology, a virtual case study entitled as "production of a ball valve for general use" has been carried out to illustrate the key activities related to setting up breeding environment, setting up & operating VE and dissolution of VE.

In the case of research, it was soon found that an unlimited number of VEs can be set up once the existing competencies have been identified. The breeding environment approach represents a very low cost solution for VE, since its only costs are labor-related.

Task decomposition is crucial both in coordinating project, providing the directions to be followed and the actions to be taken at the VE level as a whole. It is also evident from the case that contracts should be signed using digital signatures to maintain the virtuality.

Finally, the feasibility of the methodology used for the establishment of VEs has been proved effectively, as demonstrated by the case study.

The findings in the research have been validated throughout the research through the various activities as meetings, conferences, presentations and publication of journals.

8.2 Summary of contributions

In the following Table 8.1 the contributions of the dissertation are listed for each research question. As can be seen from the table, the contributions in relation to all research questions start from scratch.

Table 8.1 Summary of contributions

Research Question	**Contributions (refinements and own developments)**
BQ: Why VE?	<u>Literature survey</u> • Being rooted in established theory one of the essential activities of the research has been to conduct a literature study establishing the basis ground upon which the research should be based. The literature study has not only been conducted initially, but was extended throughout the research. Thus, a final literature review was made after the completion of research before writing up this dissertation. • Accordingly, the theoretical elements included in chapter 2 & chapter 3 more broadly covers the

	literature as well as includes most recent references compared to the appended publications.
RQ1: What is a VE?	<u>VE business model</u> • Refinement of the concept in relation to classifying different enterprise networks and virtual enterprise (ref. Section 4.2). • Concept illustration (ref. Figure 4.1) • Refinement of the life cycle model of dynamic virtual enterprise (ref. Section 4.3)
RQ2: What framework is needed to structure the body of knowledge related to VEs?	<u>ICT reference architecture</u> • Creation of the layered ICT reference architecture for distributed manufacturing systems (ref. Section 5.1) <u>Structured methodology</u> • Establishment of a structured methodology (ref. Section 5.2)
RQ3: How to prepare, set up, and operate VEs?	<u>ICT architecture</u> • Creation of the layered ICT architecture for VE system based upon the ICT reference architecture (ref. Section 6.1) <u>VE methodology</u> • Complete development of the methodology (ref. Section 6.2) ▪ Creation of IDEF0, use cases, and activity diagrams in the scope of analysis phase (ref. Section 6.2.1) ▪ Creation of class, sequence, data diagrams, and site maps in the context of design phase (ref. Section 6.2.2) ▪ Creation of system database and web pages (ref. Section 6.2.3) <u>Developed algorithms</u>

	• Analytic hierarchy process (AHP) model: Developed to contribute to the selection of the partner companies in the virtual enterprises (ref. Section 6.3). • Artificial neural network (ANN) model: Developed to assess the performance of the partner companies taking into consideration the disciplinary and quality aspects (ref. Section 6.4). • PERT analysis: Developed to calculate the probability of completing the project by the given time period (ref. Section 6.5).
RQ4: How to test and validate the developed VEM?	<u>Virtual case study</u> • Establishment of a virtual case study as "production of ball valves for general use" (ref. Section 7.1) <u>Validation platform</u> • Building a validation platform (ref. Section 7.2)

8.3 Research limitations

This methodology will enable SMEs mainly working in the mechanical sector to cooperate in a virtual and distributed environment to dramatically save time and reduce production costs. This however does not mean that the results cannot be applied in other types of sectors so easily.

Generally speaking the virtual enterprise concept have been introduced as a business concept suitable in the new economy facing the business challenges such as globalization, focus on core competencies, and customization as addressed in section 3.1. Thus, the concept has emerged as an answer to enterprises that operate in a dynamic environment compared to a conventional production networks.

Thus, the basic idea of having an enterprise network as a breeding environment for preparing and setting up projects in the form of virtual enterprises is not applicable for all types of industries. This is because most companies are engaged in a type of network in which it cooperates with other enterprises. Therefore, in the enterprise network concept, in which a degree of preparedness can be established in advance, is of general applicability. With regards to the virtual enterprise aspect of the business model it will be most applicable for enterprises engaged in projects carried out in a corporation with other partners. However, the dynamic and temporary aspects of virtual enterprise would be less relevant for enterprises that operate in stable production networks with stable product portfolio and with few or no inter-enterprise collaboration.

Thus, the findings of this research are relevant for enterprises that now and then modify their product portfolio and in doing this cooperate with other partners as part of their operations, e.g. during design, manufacturing, and service of their products.

8.4 Future work

This dissertation have addressed why, what and how questions in relation to virtual enterprises. However, this does not mean that all aspects in relation to virtual enterprises have been addressed completely leaving no questions left for future research. On the contrary this research has proved that the task of preparing, setting up and operating virtual enterprises is a complex and comprehensive task building upon several disciplines. The results can be seen as a first step towards the realization of virtual enterprises.

A realization of the vision requires a lot of further development including work related to the results of this dissertation. Especially ICT architecture and VEM could be further refined and enhanced, whereas the VE concept is at a more mature level. ICT is progressing very fast. The technology that belongs to 2-3 years ago is considered to be “old”. In such a fast paced environment, every work should be updated and continued to be more advanced. VEM would benefit from being tested further and refined based upon experiences. This could include test in different environments pursuing to refine it to a set of possible variations of the VEM relating to different types of settings distinguished by use of the contingency factors.

The main future work for this study is to keep it updated with the emerging technologies. Further refinement and elaboration of the components or technologies would increase the applicability and hence its diffusion in general.

Some other recommendations and directions for the future work are as follows:

1. Some effective tools as cost estimation & real-time monitoring modules can be integrated to the system to increase the robustness of the system.
2. Various seminars can be arranged to disseminate the usage of e-signature to partner SMEs, so that contracts can be signed over the internet using e-signatures without making doubt to maintain the virtuality.
3. A methodological approach for testing and debugging of developed software can be introduced to the methodology.
4. The design of the web pages can be enforced with visual interface design methodologies.
5. Error handling procedures for unexpected cases in the system can be developed to make the developed systems more faults tolerant.

REFERENCES

Adams, T.D., 2004, "Workflow Interoperability Between Businesses", In: Lawerence P. (ed.) WfMC Workflow Handbook. John Wiley & Sons, New York, pp. 211-221.

Afsarmanesh, H., Garita, C., Hertzberg, L.O., and Santos-Silva V., 1997, “Management of distributed information in virtual enterprises – the PRODNET approach” in Pawar, K.S. (ed.) Proceeding of the 4th International Conference on Concurrent Enterprising ICE’97, University of Nottingham: UK, pp. 241-255

Ahokangas, P., 2004, “Internationalisation and Resources”, An Analysis of Processes in Nordic SMEs, Acta Wasaensia

Alzaga, A., and Martin, J., 2004, “A Design Process Model to Support Concurrent Project Development in Networks of SMEs, in Infrastructures for Virtual Enterprises”. Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'04), Porto, Portugal, pp. 307-318.

Amaitik, S.M., 2005, “Development of a STEP Feature-based Intelligent Process Planning System for Prismatic Parts”, Ph.D. Thesis, METU

Amit, R., and Schoemaker, P.J.H., 1993, “Strategic Assets and Organizational Rent”, Strategic Management Journal, Vol. 14, pp. 33-46

Bagshaw, R.W., and Newman, S.T., 2000, "Production Code Generation and Manufacturing Analysis within a Holonically Enhanced Small Manufacturing Enterprise," Proceedings of the 10th International Conference on Flexible Automation and Intelligent Manufacturing, College Park, Maryland, pp. 1056-1065, Jun. 26-28.

Banahan, E., and Banti, M., 1999, "Report on Workshop on Legal Aspects of Virtual Organisations", Brussels, http://www.ispo.cec.be/serist/

Barnett, W., Presley, A., Johnson, M., and Liles, D.H., 1994, "An Architecture for the Virtual Enterprise" in Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, October, pp. 506-511

Barney, J., 1991, "Firm Resources and Sustained Competitive Advantage", Journal of Management, Vol. 17, No. 1, pp. 99-120

Benon-Davies, P., 1991, "Relational Database Systems: A Pragmatic Approach", Information Systems Series, Blackwell Scientific Publications, Oxford, GB

Berg, R.J., and Lieshout, J.M., 2001, "Finding symbolons for cyberspace: addressing the issue of trust in electronic commerce", Production Planning and Control 2001, Vol. 12, No. 5, pp. 514-24

Berg, R.J., and Tølle, M., 2001, "Assessing ability to execute in Virtual Enterprises", in Mo, J. & Nemes, L. (Eds.) Global Engineering, Manufacturing and Enterprise Networks, Kluwer Academic Publishers: Dordrecht, pp. 38-45

Bernus, P., Baltrusch, R., Vesterager, J. and Tølle, M., 2002, "Fast tracking ICT infrastructure requirements and design, based on Enterprise Reference Architecture and matching Reference Models", in Camarinha-Matos, L.M. (Ed.) Collaborative business ecosystems and Virtual Enterprises, Kluwer Academic Publishers: Dordrecht, pp. 293-302

Billington, C., 2004, "Strategic Supply Chain Management", OR/MS Today, pp. 20-27.

Booch, G., Rumbaugh, J., and Jacobson, I., 1999, "The Unified Modeling language user guide", The Addison-Wesley Object Technology Series. Addison-Wesley, U.S.A.

Bourdeau, M., Kazi, S.K., Rezgui, Y., and Vakola, M., 2001, "An Open System for Inter Enterprise Information Management in Dynamic Virtual Enterprises", Constructions Informatics Digital Library, http://itc.scix.net

Britton, C., and Doake, J., 2005, "Object-oriented systems development, a gentle introduction", McGraw-Hill, London

Browne, J., 1995, "The extended enterprise – manufacturing and the value chain", in Camarinha-Matos, L. & Afsarmanesh, H. (Eds.) Balanced Automation System – Architecture and Design Methods, Chapman & Hall, pp. 5-16

Browne, J., Sackett, P.J., and Wortmann, J.C., 1995, "Future manufacturing systems – Towards the extended enterprise", Computers in Industry, Vol. 25, No. 3, pp. 235-254

Brucker, P., 1999, "Resource-constrained project scheduling: notation, classification, models, and methods", European Journal of Operational Research; 112:3–41.

Byrne, J.A., Brandt, R., and Port, O., 1993, "The virtual corporation", Business Week, February 8, pp. 36-40

Camarinha-Matos, L. M., and Afsarmanesh, H., 1999a, "Infrastructures for Virtual Enterrises. Networking Industrial Enterprises", IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, 27-28 October 1999, Boston: Kluwer Academic Publishers

Camarinha-Matos, L.M., and Afsarmanesh, H., 1999b, "The PRODNET Demonstrator" in Infrastructures for Virtual Enterrises. Networking Industrial

Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, Boston: pp. 279-290

Camarinha-Matos, L.M., and Afsarmanesh, H., 1999c, “The Virtual Enterprise Concept” in Infrastructures for Virtual Enterrises. Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, Boston: pp. 3-30.

Camarinha-Matos, L.M., and Cardoso, T., 1999d, “Selection of partners for a virtual enterprise”, In: Camarinha-Matos LM, Afsarmanesh H, editors. Pro-VE’99 book, infrastructure for virtualenterprises: networking industrialenterprise s. Boston Dordrecht London: Kluwer Academic Publishers, pp. 259–78.

Camarinha-Matos, L.M., and Afsarmanesh, H., 2003, “Elements of a base VE Infrastructure”, to appear in J. Computers in Industry, Vol. 51, Issue 2,, pp. 139-163

Chapman, S.N., and Carter, P.L., 1990, “Supplier/customer inventory relationships under just-in-time”, Decision Sciences 21 (1), 35–51.

Chen, P.P.S., 1976, “The entity-relationship towards a unified view of data”, ACM Transactions on Database Systems, Vol. 1, No.2, pp. 9-36

Chen, S.J., and Hwang, C.L., 1992, “Fuzzy Multiple Attribute Decision Making—Methods and Applications”, Springer, Berlin Heidelberg.

Chen, Y., and Liang, M., 2000, “Design and implementation of a collaborative engineering information system for allied concurrent engineering”, International Journal of Computer Integrated Manufacturing, Vol. 13, No. 1, pp. 11-30

Chesbrough, H.W., and Teece, D.J., 1996, “When is Virtual Virtuous? Organizing for Innovation”, Harvard Business Review, January-February, pp. 65-73

Coad, P., and Yourdan, E., 1990, “Object oriented analysis”, 2nd Edition, Yourdan Press, USA

Collett, S., 1999, “Integration software links XML to supply chain”. Computerworld, 33 (11), 68±73.

Colquhoun, G.J., Baines, R.W., and Crossley, R., 1993, “A state of the art review of IDEF0”, International Journal of Computer Integrated Manufacturing, Vol.6, No.4, pp. 252-264

Cost, R., 1998, “Jackal: A Java-based Tool for Agent Development” Working Notes of the Workshop on Tools for Developing Agents, WS-98-10, AAAI TR 73-82.

Crossflow Project, 1999, “ESPRIT Project 28635 Cross-Organisational Workflow Management”, The Crossflow Consortium, http://www.crossflow.org

Cselenyi, J., and Toth, T., 1998, "Some Questions of Logistics in the Case of Holonic Production Systems," Journal of Intelligent Manufacturing, vol. 9, no. 2, pp. 113-118.

Davenport, T.H., and Short, J.E., 1990, “The New Industrial Engineering: Information Technology and Business Process Redesign”¸ Sloan Management Review, Summer, pp. 11-27

Davidow, W.H., 2000, “Altered States”, Forbes ASAP, May 29

Davidow, W.H., and Malone, M., 1992, “The Virtual Corporation”, New York, HarperBusiness

Debenham, H., 1998, "An Experimental Agent-based Workflow System". Proceedings of the Third International Conference on the Practical Application of Agent Technology (PAAM98), London, UK, 1998. The Practical Application

Company Ltd, pp 101-109.

Dell, M., 1999, “Direct from Dell: strategies that revolutionized an industry”, (1st ed.). New York: HarperBusiness.

Desbarats, G., 1999, “The innovation supply chain”, Supply Chain Management, 4 (1), 7±10.

Devor, R., Graves, R., and Mills, J.J., 1997, "Agile Manufacturing Research: Accomplishments and Opportunities," IIE Trans., vol. 29, no. 10, pp. 813-823.

Dhar, R., and Glazer, R., 2003, “Hedging Customers”, Harward Business Review, May, pp. 86-92

Dickson, G., 1966, “An analysis of vendor selection systems and decisions”, Journal of Purchasing 2, 28–41.

Dijkstra, E.W., 1990, “Structured Programming”, Software Engineering Techniques, NATO Science Committee, Brussels

Doz, Y.L., and Hamel, G., 2004, “Alliance Advantage. The Art of Creating Value through Partnering”, Boston: Harvard Business School Press.

Drucker, P.F., 1999, “Management Challenges for the 21st Century”, Butterworth Heinemann:Oxford

Dubois, D., Fargier, H., and Galvagonon, V., 2003, “On latest starting times and .oats in activity networks with ill-known durations”, European Journal of Operational Research 147, 266–280.

Ebay Homepage, http://www.ebay.com, last accessed: June, 2006

Elmaghraby, S.E., 1977, "Activity networks-project planning and control by network models", New York: Wiley.

Ellram, L.M., and Cooper, M.C., 1990, "Supply chain management, partnership and the shipper third party relationship". The International Journal of Logistics Management, 1 (2), 1±10.

ENSR Enterprise Survey 2003, http://www.europa.eu.int/comm/enterprise, last accessed: June 2006

Eriksson, H.E., and Penker, M., 2000, "Business modeling with UML: business patterns at work", John Wiley & Sons, New York, U.S.A.

Farrell, D., 2003, "The Real New Economy", Harward Business Review, October, pp. 104-112

Fatemi Ghomi, S.M.T., and Rabbani, M., 2003, "A new structural mechanism for reducibility of stochastic PERT networks", European Journal of Operational Research 145, 394–402. European Journal of Operational Research 109, 299–309.

Fielding, R., 1998, "Support for the Virtual Enterprise: web-based Development of Complex Information Products" Communications of the ACM, Vol. 41, No 8, pp. 84-92

Filos, E., and Banahan, E., 2001,"Will the organisation disappear? The challenges of the new economy and future perspectives" in Camarinha-Matos, L.M., Afsarmanesh, H. & Rabelo, R.J. (Eds.) E-business and Virtual Enterprises – Managing Business-to-Business Cooperation, Kluwer Academic Publishers: Dordrecht, pp. 3-20

Flynn, D., 2003, "Information Systems Requirements: Determination and Analysis", 2nd Edition, Mc Graw-Hill, London

Forsythe, C., 1997, "Human Factors in Agile Manufacturing: A Brief Overview with

Emphasis on Communications and Information Infrastructure," Human Factors and Ergonomics in Manufacturing, vol. 7, no. 1, pp. 3-10.

Frederix, F:L.M., 1998, "Agility and Human Factors in the Virtual Enterprise, in Innovation, Agility, and the Virtual Enterprise", Proceedings of the 10th International IFIP WG 5.2/5.3 Conference, PROLOMAT 98, Trento, Italy, pp. 737-748

GENIAL Project Homepage and Papers, http://cic.cstb.fr/ilc/publicat/list.htm or http://www.clab. de/genial/index.html, last accessed: June 2006

Georgakopoulos, D., 1998, "Collaboration Management Infrastructure for Comprehensive Process and Service Management" International Symposium on Advanced Database Support for Workflow Management, Enschede, The Netherlands, May 19th.

Geppert, A., Kradolfer, M., and Tombros, D., 1998, "Market-Based Workflow Management" Int'l IFIP Conf.on Distributed Systems for Electronic Commerce, Hamburg, Germany, June 1998

GLOBEMEN, "Global Engineering and Manufacturing in Enterprise Networks", IMS 99004, IST- 1999-60002, 2000-2002, URL: http://globemen.vtt.fi, last accessed: June 2006

Goldman, S.L., Nagel, R.N., and Preiss, K., 1995, Agile competitors and virtual organizations:Strategies for Enriching the Customer, New York, Van Nostrand Reinhold

Goranson, H.T., 1999, "The agile virtual enterprise: cases, metrics, tools", Westport, CT: Quorum Books.

Gray, C.F., and Larson, E.W., 2000, "Project Management – The Managerial

Process", McGraw-Hill.

Grefen, J., Pernini, B., and Sanchez G., 1999, "Database support for workflow management: The WIDE project" Kluwer Academic Publishers, ISB7923-8414-8

Gulati, R., and Garino, J., 2000, "Getting the Right Mix of Bricks & Clicks", Harvard Business Review, May-June, pp. 107-114

Gunasekaran, A., 1998, "Agile Manufacturing: Enablers and an Implementation Framework," International Journal of Production Research, vol. 36, no. 5, pp. 1223-1247.

Haeckel, S.H., and Nolan, R.L., 1993, "Managing by wire", Harvard Business Review, September-October, pp.122-132

Hakansson, H., and Snehota, I., 2005, "Developing Business Relationships". London, Routledge.

Hammel, G., Doz, Y.L., and Prahalad, C.K., 1989, "Collaborate with Your Competitors – and Win", Harvard Business Review, January-February, pp. 133-139

Hammel, G., 1996, "Strategy as revolution", Harvard Business Review, July-August, pp.69-82

Hammel, G., 1999, "Bringing Silicon Valley Inside", Harvard Business Review, September- October, pp.71-84

Hammer, M., 1990, "Reengineering Work: Don't Automate, Obliterate", Harvard Business Review, July-August, pp. 104-112

Hammer, M., 2000, "The Rise of the Virtual Enterprise", Information Week, 20 March, p. 152

Handy, C., 1995, “Trust and the Virtual Organization”, Harvard Business Review, May-June, pp.40-50

Harmon, P., and Sawyer, B., 2004, “UML for Visual Basic Developers”, Morgan Kaufmann Publishers, San Francisco

Hartel, I., and Burger, G., 2001, “Virtual Service Enterprise: A model for virtual collaboration in after-sales service in the one-of-a-kind industry”, in Ferroni, B., Boer, C.R. & Anderson, C.M. (Eds.) International IMS Project Forum 2001, A common platform for cross-fertilization among IMS projects and promotion of the IMS program, ISBN 88-900298-3-8, pp. 255-262

Haykin, S., 1994, “Neural networks, a comprehensive foundation”, New York: Macmillan

Hartel, I., and Burger, G., 2001, “Virtual Service Enterprise: A model for virtual collaboration in after-sales service in the one-of-a-kind industry”, in Ferroni, B., Boer, C.R. & Anderson, C.M. (Eds.) International IMS Project Forum 2001, A common platform for cross-fertilization among IMS projects and promotion of the IMS program, ISBN 88-900298-3-8, pp. 255-262

Hernandez, M.J., 2001, “Database Design for Mere Mortals A hands-on Guide to Relational Database Design”, Addison-Wesley, USA

Hetzel, B., 2004, “The Complete Guide to Software Testing”, 2nd Edition, John Wiley and Sons, USA

Hoffner, Y., and Crawford, B., 1997, “Using Interception to Create Domains in Distributed Systems”. Joint International Conference on Open Distributed Processing (ICODP) and Distributed Platforms (ICDP), Toronto, Canada

Hoffner, Y., and Schade, A., 1998, "Co-operation, Contracts, Contractual Match-Making and Bindings" 2nd International Enterprise Distributed Object Computing Workshop (EDOC 98), 3- 5 November, San-Diego, USA

Hoffner, Y., 1999, "Supporting Contract Match-Making" IEEE 9th International Workshop on Research Issues on Data Engineering, RIDE-VE 99, Sydney, Australia.

Hongmei, G., Biqing, H., and Shouju, R., 2000, "A UML and Petri Nets Integrated Modeling Method for Business Processes in Virtual Enterprises", American Association for Artificial Intelligence.

Hunt, I., Caskey, K., and Browne, J., 1999, "SMEs in the Virtual Enterprise - LOGSME enables IT Support, in Infrastructures for Virtual Enterprises" Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, pp. 333-342.

IDEF0, 1993, "Integration Definition for Function Modeling", Draft Federal Information, Processing Standards Publication 183

IDEF1/X, 1993, "Integration Definition for Information Modeling", Draft Federal Information, Processing Standards Publication 184

IMS International, 2000, "IMS Annual Progress Report 2000", Tokyo: IMS International

Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, pp. 293-306.

Isaksen, A., 2002, Spesialiserte produksjonsområder I Norden (Specialized production areas in the Nordic Countries). Nordisk Samhällsgeografisk

Jagdev, H.S., and Browne, J., 1998, "The extended enterprise — a context for manufacturing”. Prod Plann Control; 9(3):216–29.

Jagdev, H.S., and Thoben, K.D., 2001, “Anatomy of enterprise collaborations”, Production Planning & Control, Vol. 12, No. 5, pp. 437-451

Jung, M., Chung, M.K., and Cho, H., 1996, "Architectural Requirements for Rapid Development of Agile Manufacturing Systems," Computers & Industrial Engineering: Proceedings of the 19th International Conference on Computers and Industrial Engineering, vol. 31, no. 3/4, pp. 551-554.

Kamath, M., and Ramamritham, K., 2002 “Pragmatic Issues in Coordination Execution and Failure Handling of Workflow in Distributed Workflow Control Architectures” Oracle Corp. Technical Report.

Katzy, B., & Schuh, G., 1998, “The Virtual Enterprise”, in Molina, A., Kusiak, A. & Sanchez, J. (Eds.) Handbook of Life Cycle Engineering - Concepts, Methods and Tools, Kluwer Academic Publishers: Dordrecht, pp. 59-91

Katzy, B., Schuh, G., & Millarg, K., 1996, Die Virtuelle Fabrik – Produzieren im Netzverk, TR Transfer, nr. 43, pp. 30-34

Kerste, R., and Muizer, A.P., 2005, “The relationship between SME co-operation and market concentration: Evidence from small retailers in the Netherlands”, Journal of Small Business Management, April, pp. 68-73.

Kovac, F.J., 1994, “Creating the Virtual Enterprise”, Chemtech, Vol. 24, November, pp. 7-11

Koestler, A., 1967, “The Ghost in the Machine”. New York: The Macmillan Company.

Lechner, C., and Dowling, M., 2003, “Firm networks: external relationship as sources for the growth and competitiveness of entrepreneurial firms”. Entrepreneurship & Regional Development, 15, pp. 1-26.

Lee, G.H., 1998, "Designs of Components and Manufacturing Systems for Agile Manufacturing," International Journal of Production Research, vol. 36, no. 4, pp. 1023-1044.

Lee, W.B., and Lau, H.C.W., 1999, “Factory on demand: the shaping of an agile production network”, International Journal of Agile Management Systems, 1 (2), 83±87.

MacCarthy, B.L., Liu, J., 1993, “A new classification scheme for flexible manufacturing systems”, International Journal of Production Research, Vol. 31, No.2, pp. 299-309

Malone, T., and Crowston, K., 1991, “Towards an interdisciplinary theory of coordination”. Technical Report CCS RT #120 SS WP#3294-91-MSA, Massachusetts Institute of Technology (MIT)

Malone, T.W., & Laubacher, R.J., 1998, “The dawn of the e-lance economy”, Harvard Business Review, September-October, pp. 145-152

Malone, T.W., and Rockart, J.F., 1991, Computers, Networks, and the Corporation. Scientific American vol. 265, no. 3. pp. 128-136.

Maloni, M.J., and Benton, W.C., 1997, ”Supply chain partnership: opportunities for operations research”, European Journal of Operational Research;101:419–29.

Marvelous Project Home Page, http://www.biba.uni-bremen.de/projects/marvelous, last accessed: June 2006

Mazzeschi, M., 2001, “The Virtual Organisation”, in Thoben, K-D., Weber, F. &

Pawar, K.S. (Eds.) Proc. of the 7th International Conference on Concurrent Enterprising: ‘Engineering the Knowledge Economy through Co-operation’, 27-29 June (ICE2001), Bremen, pp. 331-335

McCaffer, R., and Garas, F., 1999, “eLSEwise: European Large Scale Engineering Wide Integration Support Effort, Engineering Construction and Architectural Management,” Special Issue, 6 (1), ISSN 0969 9988.

McCormack, R., 1997, "Biological Concepts Make Their Way into Manufacturing," Manufacturing News, vol. 4, no. 27.

McGrath, G.M., & More, E., 2003, “Strategic Partnership in the IT Sector: Trust, Power and Inter-Organizational Collaboration”, Proc. 12th Australian Conference on Information Systems, pp. 586-596

Microsoft white paper, Microsoft Windows DNA architecture, www.microsoft.com/dna , last accessed: June 2006

Microsoft white paper, Microsoft .NET framework, www.microsoft.com/net, last accessed: June 2006

Miles, R.E., and Snow, C.C., 1984, “Fit failure and the hall of fame”, California Management Review, Vol. 26, No. 3, pp. 10-28

Miles, R.E., and Snow, C.C., 1986, “Organizations: New Concepts for New Forms”, California Management Review, Vol. 28, No. 3, pp. 62-73

Miles, R.E., and Snow, C.C., 1992, “Causes of Failure in Network Organizations”, California Management Review, Vol. 34, No. 4, pp. 53-72

Miles, R.E., Snow, C.C., and Miles, G., 2000, “TheFuture.org”, Long Range Planning, Vol. 33, pp. 300-321

Mitrovic, D., Hunter, I., and Male, S., 2003, "Characteristics of Networked Enterprise in Global Construction" in Proceedings of ICE'03, International Conference on Concurrent Enterprising, The Hague, The Netherlands, 15-17 March 2003, Nottingham: University of Nottingham, ISBN 0 9519759 86 pp. 447-454.

Mohan, K., and Krithi, R., 1998, "Pragmatic Issues in Coordination Execution and Failure Handling of Workflow in Distributed Workflow Control Architectures", Oracle Corp. Technical Report

Moller, C., 1997, "The virtual organisation", Automation in construction, Vol. 6, pp. 39-43

Morgan, K., 2002, "The Learning Region: Institutions, Innovations and Regional Renewal", In Regional Studies, Journal and the Regional Studies of Association. Vol. 3, No. 5.

Mowshowitz, A., 1994, "Virtual Organization: A Vision of Management in the Information Age", The Information Society, Vol. 10, pp. 267-288

Mowshowitz, A., 1997, "Virtual Organization", Communication of the ACM, September, Vol. 40, No. 9, pp. 30- 37

Mowshowitz, A., 2001, "Virtual Organization: The New Feudalism", Computer, April, pp. 110-112

Mowshowitz, A., 2002, "Virtual Organization, Toward a Theory of Societal Transformation Stimulated by Information Technology", Quorum Books: Westport

"NIIIP Concepts", National Industrial Information Infrastructure Protocols (NIIIP) Consortium, December, URL: http://www.niiip.org, last accessed: June 2006

"NIIIP Reference Architecture", available at http://www.niiip.org, last accessed: June 2006

Noori, H., and Mavaddat, F., 1998, "Enterprise Integration: Issues and Methods," International Journal of Production Research, vol. 36, no. 8, pp. 2083-2097.

Noran, O., 2000, "VITE – The Fruit and Vegetable Virtual Transport Enterprise", School of CIT, Griffith University. www.cit.gu.edu.au/~noran

Nwana, H., Ndumu, D., Lee, L., and Collis, J., 1999, "ZEUS: A Toolkit for Building Distributed Multi- Agent Systems". In: Applied Artifical Intelligence Journal, Vol 13, Number 1, 1999. http://www.labs.bt.com/projects/agents/index.htm

Oates, D., 1998, "Outsourcing and the Virtual Organization", The Incredible Shrinking Company, Random House: London

Ollus, M., Ranta, J., and Ylä-Anttila, P., 1999, "Network Revolution; How to manage networking enterprises (in Finnish), Taloustieto, Helsinki, ISBN 951-628-282-2

OSMOS Project, http://osmos.vtt.fi/, last accessed: June 2006

Ouzounis, V., 1998a, "Electronic Commerce Commercial Scenarios, Business Models and Technologies for SME's", invited paper, European Multimedia, Microprocessor Systems and Electronic Commerce Conference and Exposition (EMMSEC 98), Bordeaux, France.

Ouzounis, V., 1998b, "An Overview of Advanced Electronic Commerce Platforms and Architectures", invited paper, Information Society Technologies (IST) 98, Vienna, Austria.

Ouzounis, V., 1998c, "Electronic Commerce and New Ways of Work – An R&D

RoadMap”, European Commission –Directoral General III

Ouzounis, V., 2000, “Managing Dynamic Virtual Enterprises using FIPA Agents“ submitted as a chapter in the book „Managing the Virtual Web Organization in the 21st Century: Issues and Challenges" authored by Ulrich Franke, Cranfield University

Pawar, K.S., and Driva, H., 2000, “Electronic trading in the supply chain: a holistic implementation framework”, Logistics Information Management, 13 (1), 21±32.

Pedersen, J.D., and Berg, R.J., 2001, ”Supporting partner selection for Virtual Enterprises”, in Mo, J. & Nemes, L. (Eds.) Global Engineering, Manufacturing and Enterprise Networks, Kluwer Academic Publishers: Dordrecht, pp.95-102

Peterson, L.L., and Davie, B.S., 2000, “Computer Networks a Systems Approach”, 2nd Edition, Morgan Kaufmann Publishers, USA

Petroni, A., and Braglia, M., 2000, “Vendor selection using principal component analysis”. Journal of Supply Chain Management 36 (2), 63–69.

Plonka, F.E., 1997, "Developing a Lean and Agile Workforce," Human Factors and Ergonomics in Manufacturing”, vol. 7, no. 1, pp. 11-20.

Prahalad, C.K., and Hamel, G., 1990, “The Core Competence of the Corporation”, Harvard Business Review, May-June, pp. 79-91

Prahalad, C.K., and Hamel, G., 1994, “Strategy as a field of study: Why search for a new paradigm?”, Strategic Management Journal, Vol. 15, pp. 5-16

Preiss, K., Goldman, S.L., and Nagel, R.N., 1996, “Cooperate to Compete, Building Agile Business Relationships”, Van Nostrand Reinhold: New York

Porter, M.E., 2001, “Strategy and the Internet”, Harvard Business Review, March, pp. 62-78

Prodnet II Project, www.uninova.pt/~prodnet, last accessed: June 2006

Purdy, L., and Safayeni, F., 2000, “Strategies for supplier selection: A framework for potential advantages and limitations”. IEEE Transactions on Engineering Management 47 (4), 435–443.

Rakitin, S.R., 2002, “Software Verification and Validation, A Practitioner’s Guide”, Artech House, Boston

Ramakrishnan, R., 2001, “Database Management Systems”, McGraw-Hill, MA

Rational Software Co., 1997, “UML Notation Guide”, www.rational.com/uml

Rayport, J.F., and Sviokla, J.J., 1994, “Managing in the Marketspace”, Harvard Business Review, November-December, pp. 141-150

Rayport, J.F., and Sviokla, J.J., 1995, “Exploiting the Virtual Value Chain”, Harvard Business Review, November-December, pp. 75-85

Reimann, M.D., and Sarkis, J., 1996, "An Infrastructure for Agile Enterprise Product Development Systems," Flexible Automation and Intelligent Manufacturing 1996: Proceedings of the Sixth International FAIM Conference, Atlanta, pp. 57-66.

Ross, D.T., 1985, “Applications and extensions of SADT”, Computer, April, pp. 25-34

Rumbaugh, J., Jocabson, I., and Booch, G., 1999, “The Unified Modeling Language Reference Manual”, Addison-Wesley, MA

Saaty, T.L., 1980, “The Analytic Hierarchy Process”, New York:McGraw-Hill

Sabherwal, R., 2003, “The role of trust in outsourced IS development projects”, Communications of the ACM, Vol. 42, No. 2, pp. 80-86

Salkari, I., Kalliokoski, P., and Poikkimäki, J., 2002, “eBusiness Readiness of SMEs - Key Results of a Regional Study in Finnish Metal Industry Networks”. Proceedings of eBusiness & eWork 2002 Conference, Prague.

Sarı, B., Amaitik, S., and Kilic, S.E., 2003, “A Neural Network Model for Selection of Machining Parameters in CAPP”, WESIC 2003 (4th Workshop on European Scientific and Industrial Collaboration), Miskolc, Hungary, pp. 355-362

Sarı, B., and Kilic, S.E., 2004, “Development of an SME-based Virtual Enterprise”, 11th International Machine Design and Production Conference, Antalya, Turkey, 79-93

Sarı, B., Kılıç S.E., and Şen, D.T., 2005a, “Dağıtık İmalat Teknolojilerinde KOBİ’lere Yönelik Sanal Fabrika Sisteminin Geliştirilmesi”, Machinery MakinaTek Aylık İmalat ve Teknoloji Kültürü Dergisi, pp. 108-112, November 2005 – pp. 108-111, December 2005.

Sarı, B., Kılıç, S.E., and Şen, D.T., 2005b, “Dağıtık İmalat Teknolojilerinde KOBİ’lere Yönelik Sanal Fabrika Sisteminin Geliştirilmesi”, TMMOB Makina Mühendisleri Odası Konya Şubesi III. Makina Tasarım ve İmalat Teknolojileri Kongresi, Konya, Turkey, pp. 65-75.

Sarı, B., Kılıç, S.E., and Şen, D.T., 2006a, “Formation of Dynamic Virtual Enterprises and Enterprise Networks”, International Journal of Advanced Manufacturing Technology, ACCEPTED

Sarı, B., Amaitik, S.M, and Kılıç, S.E., 2006b, “A Neural Network Model for the

Assessment of Partners' Performance in Virtual Enterprises", International Journal of Advanced Manufacturing Technology, DOI 10.1007/s00170-006-0642-z.

Sarkis, J., and Talluri, S., 2002, "A model for strategic supplier selection". Journal of Supply Chain Management 38 (1), 18–28.

Simon, E., 1996, "Distributed Information Systems from client-server to Distributed Media", McGraw-Hill, UK

Spinosa, L.M., Rabelo, R., and Klen, A.P., 1998, "High-level Coordination of Business Processes in a Virtual Enterprise", in Innovation, Agility, and the Virtual Enterprise, Proceedings of the 10th International IFIP WG Conference, Prolomat 98, pp. 725-736

Spivey, W.A., Munson, J.M., and Wolcott, J.H., 1997, "Improving the New Product Development Process: A Fractal Paradigm for High-Technology Products," Journal of Product Innovation Management, vol. 14, no. 3, pp. 203-218.

Stevenson, W.J., 2002, "Operation Management", seventh ed. McGraw-Hill, London.

Storey, D.J., 2003, "Understanding the Small Business Sector", Routledge, London/New York.

Stricker, C., Riboni, S., Kradolfer, M., and Taylor. J., 2000, "Market-based Workflow Management for Supply Chains of Services." In Proc. 33rd Hawaii Int'l Conference on System Sciences (HICSS-33), Maui, Hawaii.

Sun, H., and Venuvinod, P.K., 2001, "The Human Side of Holonic Manufacturing Systems", Technovation, vol. 21, no. 6, pp. 353-360.

Tharumarajah, A., Wells, A.J., and Nemes, L., 1996, "Comparison of the Bionic, Fractal and Holonic Manufacturing System Concepts," International Journal of Computer Integrated Manufacturing, vol. 9, no. 3, pp. 217-226.

Tombros, D., 1999, “An Event- and Repository-Based Component Framework for Workflow System Architecture” University of Zurich.

Tombros, D., and Geppert, A., 2000, “Building Extensible Workflow Systems using an Event-Based Infrastructure”, Proc. 12th Conf. on Advanced Information Systems Engineering, Stockholm, Sweden.

Ueda, K., 1992, "A Concept for Bionic Manufacturing Systems Based on DNA-Type Information," Human Aspects in Computer Integrated Manufacturing: Proceedings of the Eighth International PROLAMAT Conference, Tokyo, pp. 853-863, Jun. 24-26.

Upton, D.M., & McAfee, A., 1996, ”The Real Virtual Factory”, Harvard Business Review, July-August, pp. 123-133

Ünver, Ö., 2000, “A systems framework and structured methodology for design and development of manufacturing control systems using n-tier client/server technology”, Ph.D. Thesis, METU

Valckenaers, P., Van Brussel, H., Wyns, J., Bongaerts, L., and Peeters, P., 1998, "Designing Holonic Manufacturing Systems," Robotics and Computer-Integrated Manufacturing, vol. 14, no. 5/6, pp. 455-464.

Valikangas, P., Kazi, A.S., Puttonen, J., Sulkusalmi, M., and Hannus, M., 2000, “Knowledge Creation and Virtual Enterprises in Power Plant Construction”, The 4th International Conference on Design of Information Infrastructure Systems for Manufacturing 2000 (DIISM2000).

VEGA Project Homepage and Papers, http://www.fruitveg.com/uk/welcome.html, last accessed: June 2006

Veloso, M., Pollack, M., and Cox, M., 1998, "Rationale -based monitoring for planning in dynamic environments". In proceedings of Artificial Intelligence Planning Systems (AIPS-98) Pittsburg.

Venkatadri, U., Rardin, R.L., and Montreuil, B., 1997, "A Design Methodology for Fractal Layout Organization", IIE Trans., vol. 29, no. 10, pp. 911-924.

VENTO Project, http://www.cas.de/English/Research/eu_overview_ventoE.asp, last accessed: June 2006

Vesterager, J., Tuxen, J., Danielsen, B.E., and Frederiksen, T.S., 1997, "Concurrent engineering in the virtual enterprise" in Pawar, K.S. (ed.) Proceeding of the 4th International Conference on Concurrent Enterprising ICE'97, University of Nottingham: UK, pp. 363-372

Vesterager, J., Larsen, L.B., and Gobbi, C., 1999, "Architecture and methodology for creating virtual enterprises" - results from GLOBEMAN21, presented at the IMS GLOBEMAN21 Open Day, March 9th, TIME24, Tokyo, Japan

VIVE Reference Model, http://www.vive-ig.net, last accessed: June 2006

VOSTER, Virtual Organisations Cluster, IST-2001-32031, URL: http://voster.vtt.fi, last accessed: June 2006

Wallenklint, J., 2001, "A Managerial Perspective on Small-Firm Network Organizations", An Overview of the Field and a Study of Practice. Department of Business Administration and Social Science. Division of Industrial Organization, 2001:46.

Wang, D., Yung, K.L., and Ip, W.H., 2001, "A heuristic genetic algorithm for subcontractor selection in a global manufacturing environment", IEEE Trans Syst Man Cybern C; 31(2):189–98.

Wang, L., 2001, " Integrated Design-To-Control Approach for Holonic Manufacturing Systems," Robotics and Computer Integrated Manufacturing, vol. 17, no. 1-2, pp. 159-167.

Warnecke, H., 1993, "The Fractal Company: A Revolution in Corporate Culture". New York: Springer-Verlag.

Weitzel, T., Buxmann, P., and Westarp, F., 1999, "A Communication Architecture for the Digital Economy - 21st Century EDI" In Proc. 33rd Hawaii Int'l Conference on System Sciences (HICSS-33), Maui, Hawaii.

Welter, F., and Havnes, P.A., 2000, "SMEs and environmental turbulence". In H.J. Pleitner and W. Weber (eds.) SMEs in the 21st century - impulses, perspectives, concepts. KMU HSG, St. Gallen.

Wernerfelt, B., 1984, "A Resource-based View of the Firm", Strategic Management Journal, Vol. 5, pp. 171-180

Weston, R.H., 1998, "Integration Infrastructure Requirements for Agile Manufacturing Systems," Journal of Engineering Manufacture: Proceedings of the Institution of Mechanical Engineers, Part B, vol. 212, no. B6, pp. 423-437.

Wognum, N., Thoben, K.D., and Pawar, K.S., 1999a, Proceedings of ICE'99, International Conference on Concurrent Enterprising, The Hague, The Netherlands, Nottingham: University of Nottingham, ISBN 0 9519759 86.

Wognum, P.M., and Faber, E.C.C., 1999b, "A Framework for Improving the Quality of Operation in a Virtual Enterprise" in Infrastructures for Virtual Enterrises. Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working

Conference for Virtual Enterprises (PROVE' 99), Porto, Portugal, p. 365-376.

Womack, J.P., Jones, D.T., and Roos, D., 1990, "The Machine that Changed the World". New York: Simon & Schuster, Inc.

Womack, J.P., and Jones, D.T., 1996, "Lean Thinking: Banish Waste and Create Wealth in Your Corporation". New York: Simon & Schuster, Inc.

Wortmann, J.C., 1992, "Factory of the future: towards an integrated theory for one of a kind production". In: Hirsch BE, Thoben KD, editors. One of a kind production: new approaches. Amsterdam: North-Holland, pp. 37–74.

Wu, N.Q., Mao, N., and Qian, Y.M., 1999, "An approach to partner selection in agile manufacturing", Journal of Intelligent Manufacturing; 10(6):519–29.

X-CITTIC, http://cordis.europa.eu/ist/98vienna/xxcittic.htm, last accessed: June 2006

X-Economy, http://www.thexeconomy.com, last accessed: June 2006

Zarli, A., and Poyet, P., 1999, "A Framework for Distributed Information Management in the Virtual Enterprise", The VEGA Project, in Infrastructures for Virtual Enterrises. Networking

Zell, A., 1995, "Stuttgart Neural Network Simulator", Software and User Manual, Version 4.0, Report No. 6/95

APPENDIX A

IDEF0 & UML DIAGRAMS

A.1 IDEF0 Diagrams

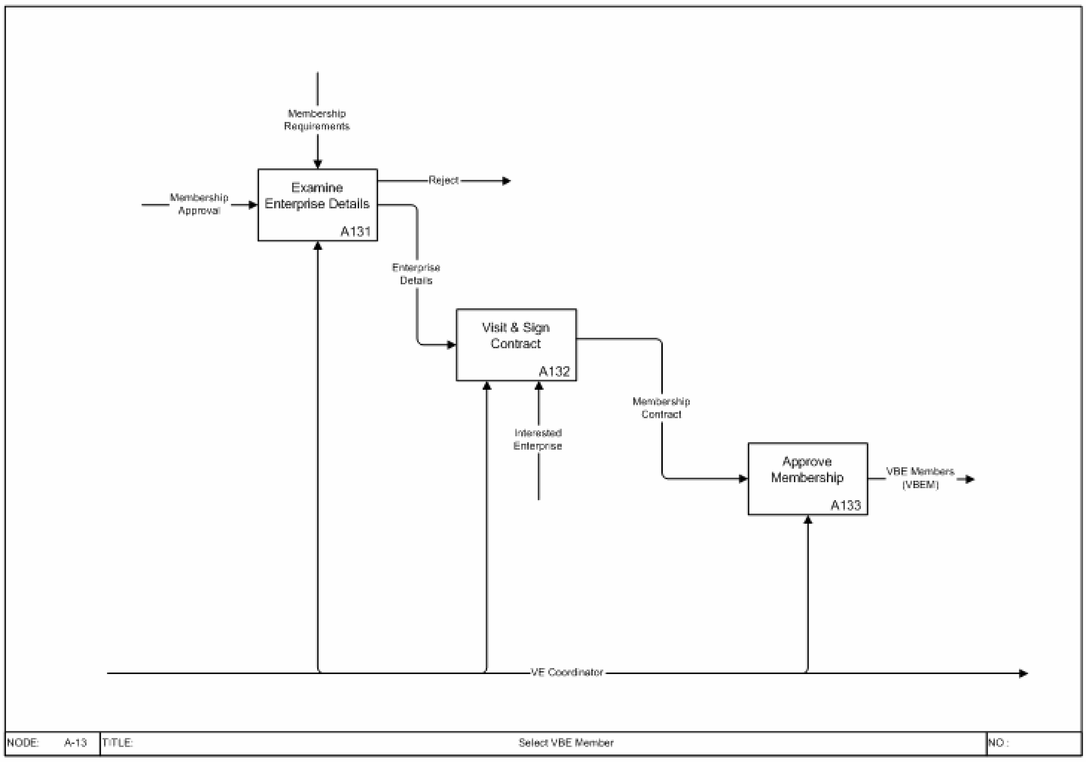


Figure A1 Decomposition of node A13

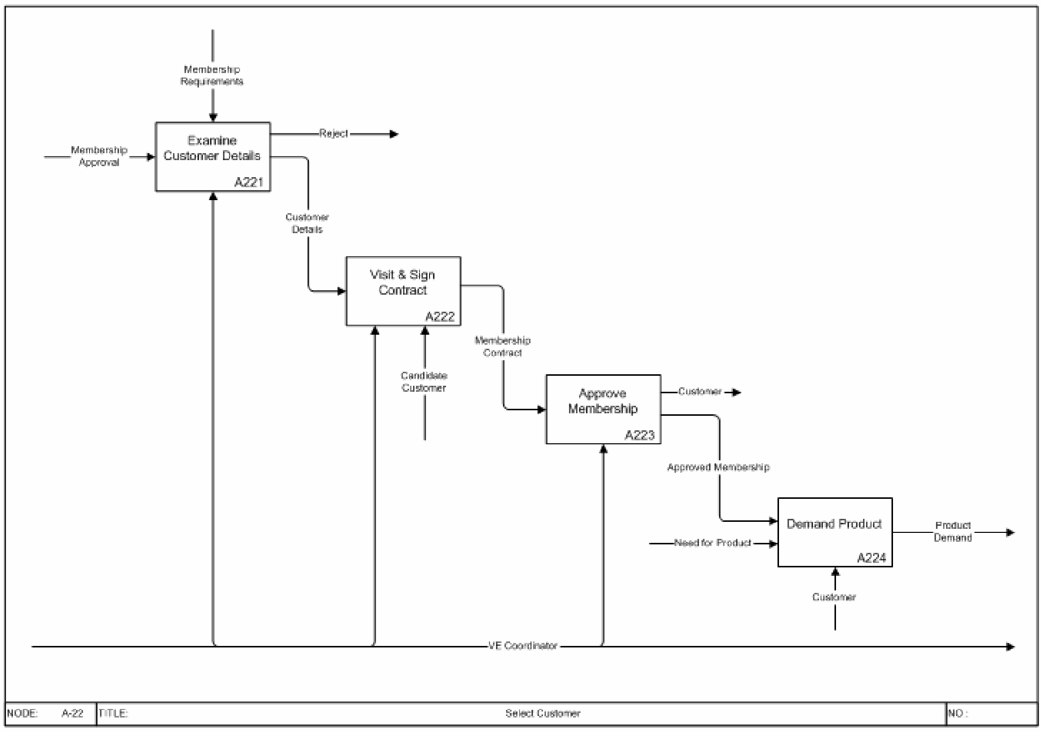


Figure A2 Decomposition of node A22

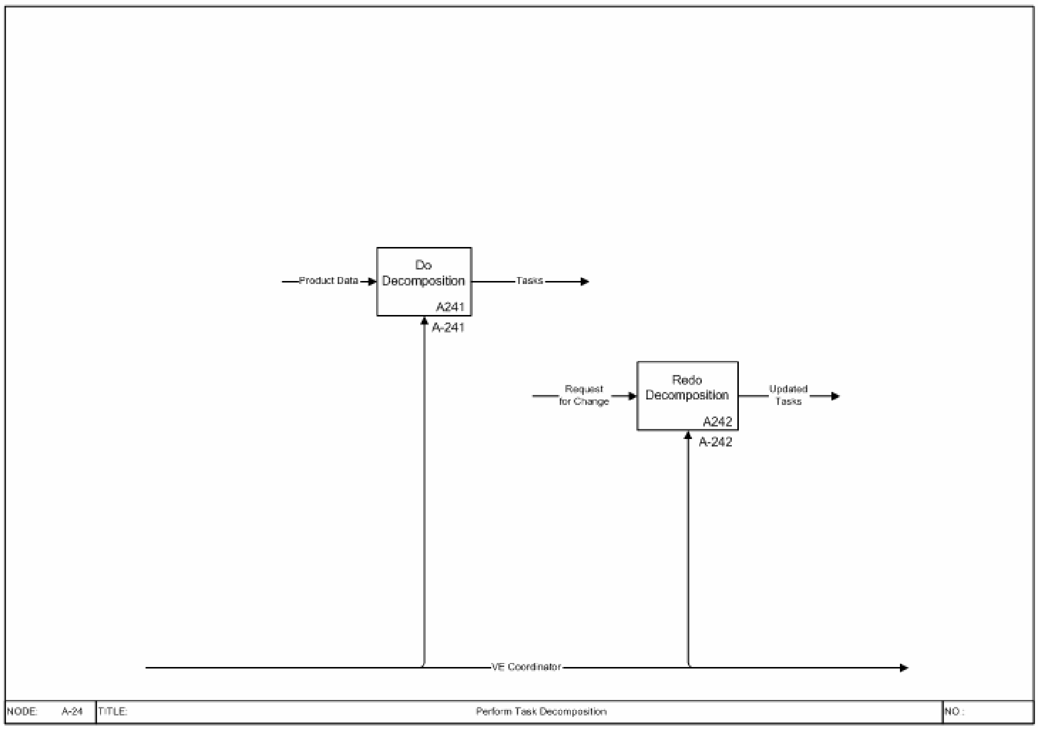


Figure A3 Decomposition of node A24

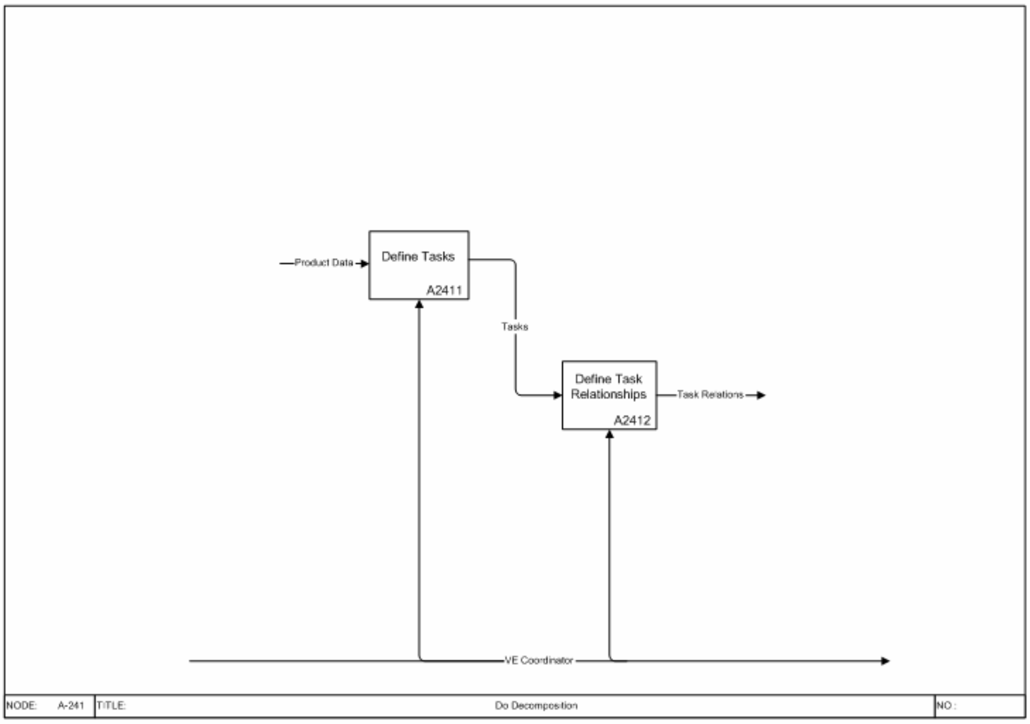


Figure A4 Decomposition of node A241

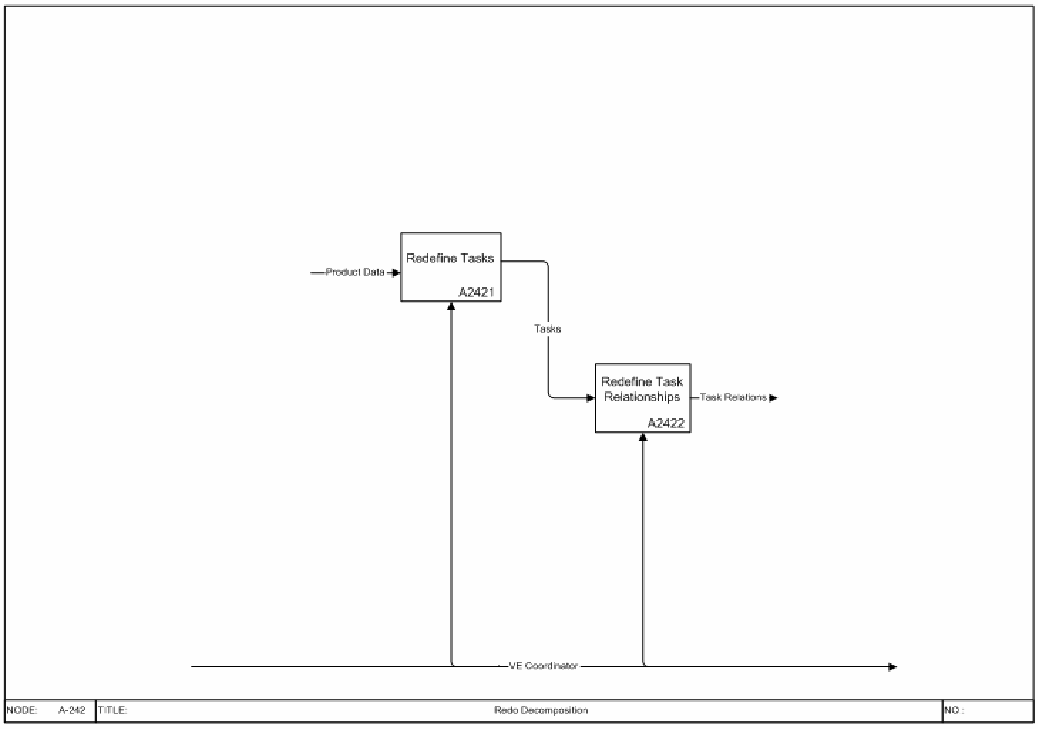


Figure A5 Decomposition of node A242

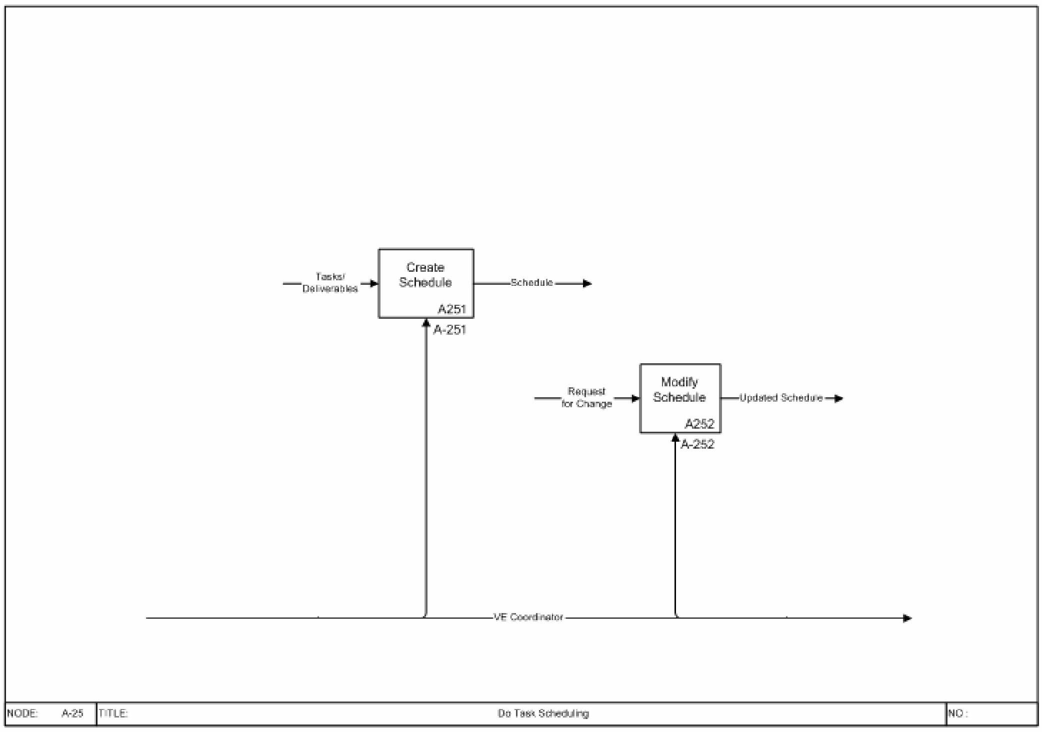


Figure A6 Decomposition of node A25

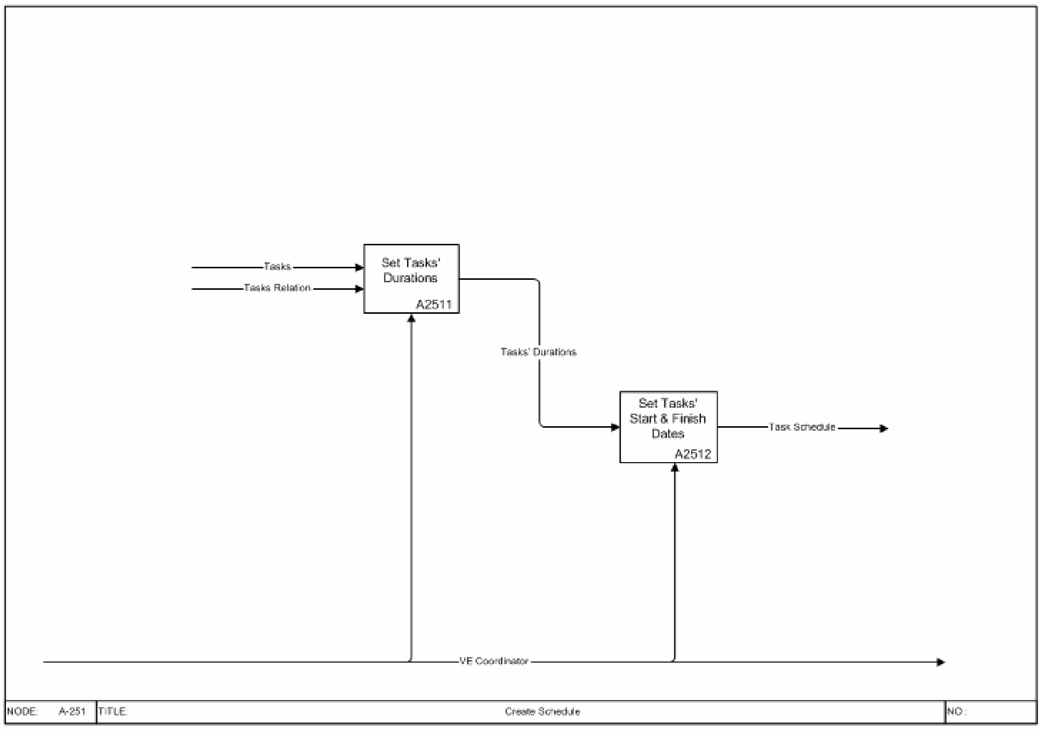


Figure A7 Decomposition of node A251

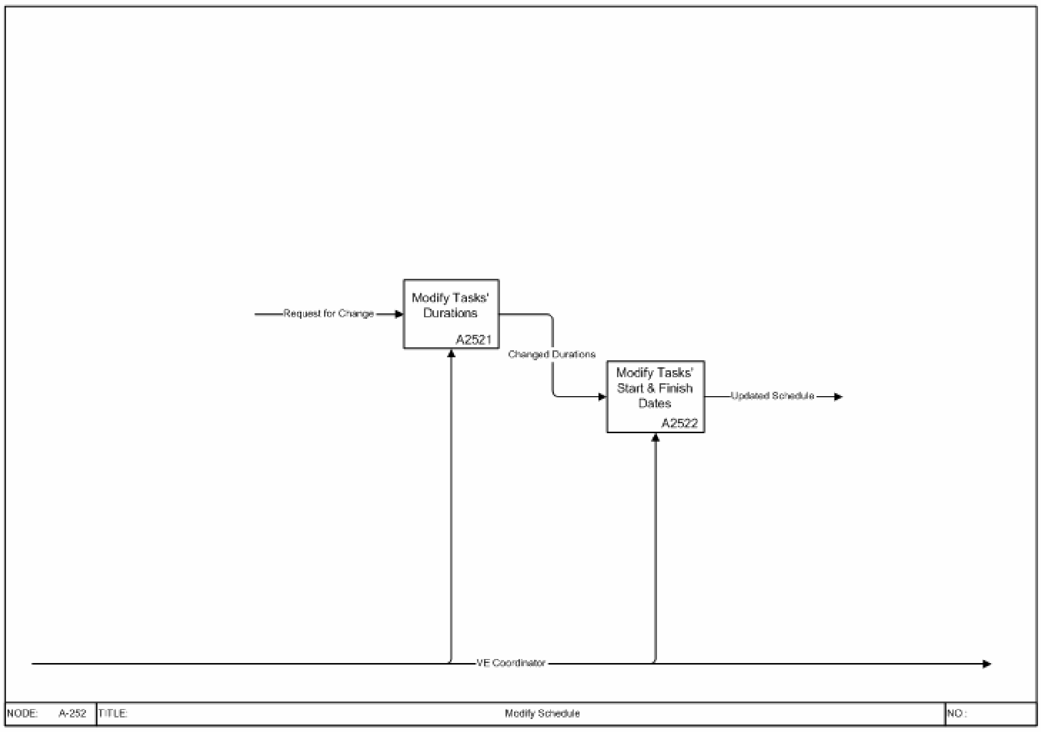


Figure A8 Decomposition of node A252

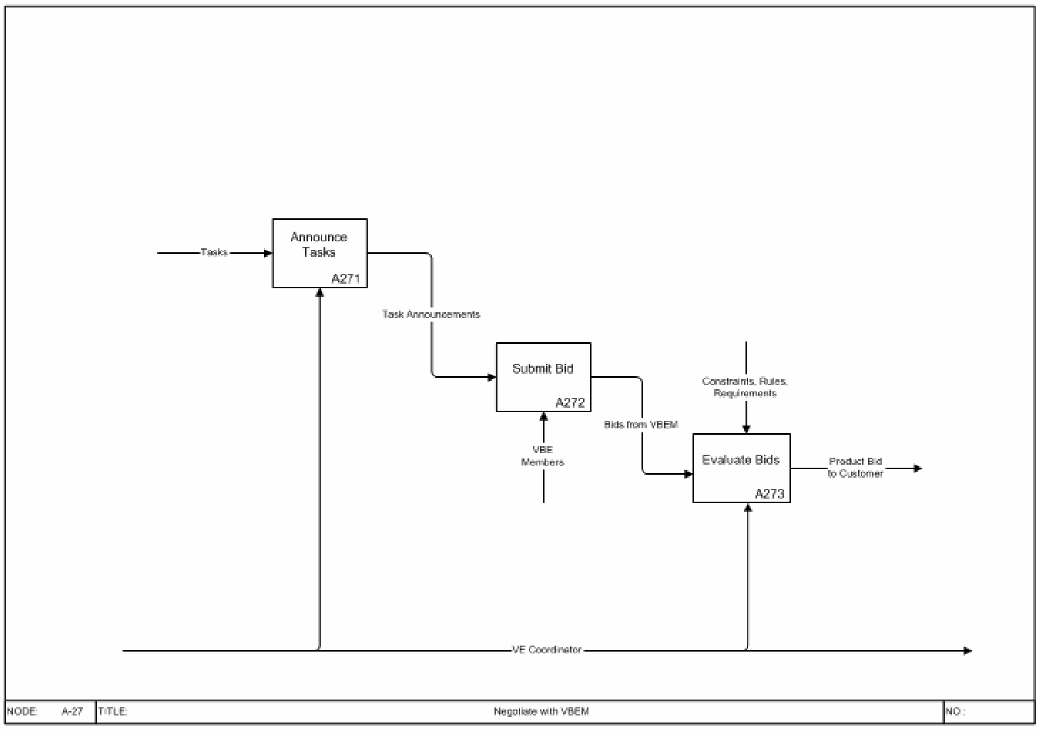


Figure A9 Decomposition of node A27

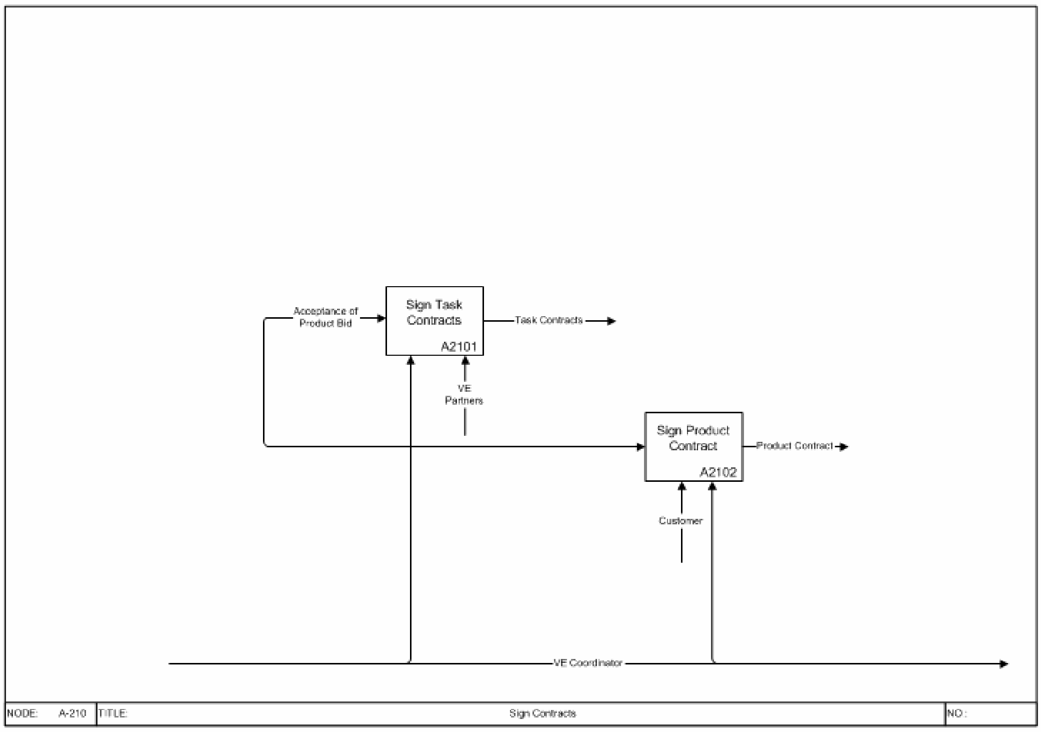


Figure A10 Decomposition of node A210

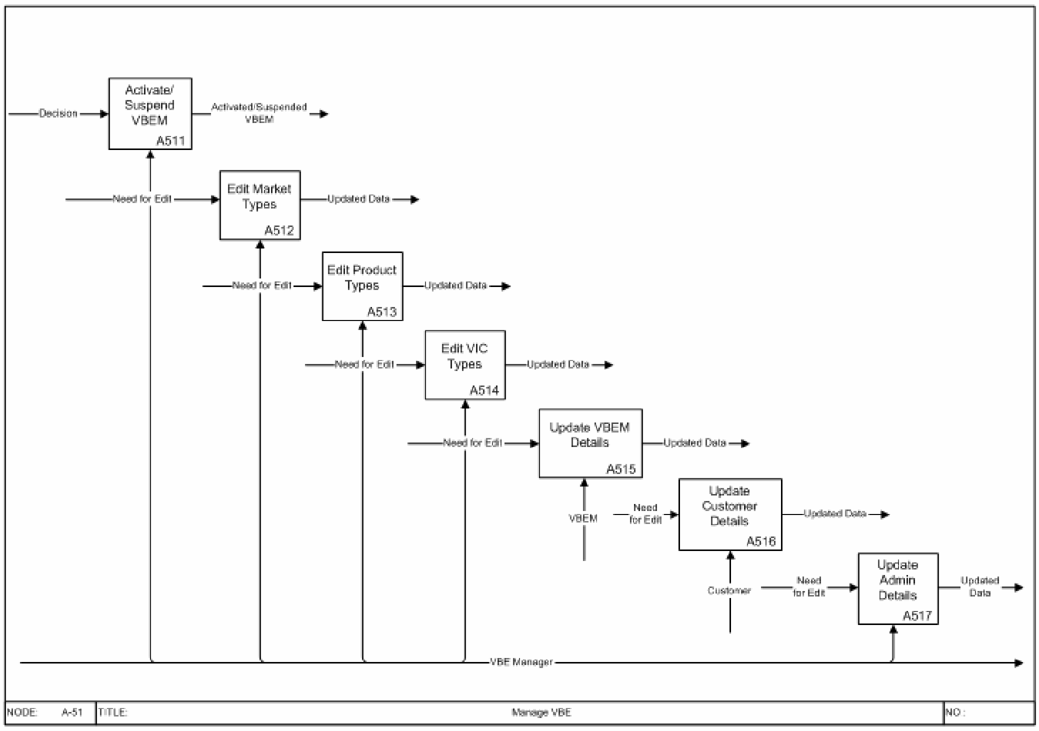


Figure A11 Decomposition of node A51

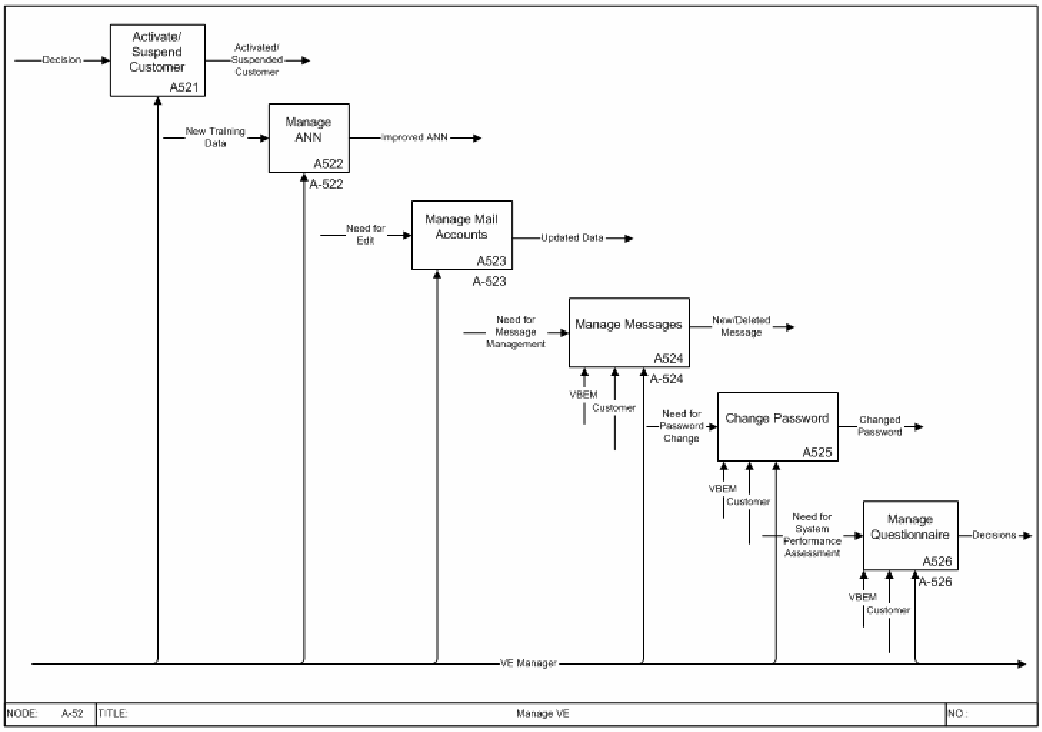


Figure A12 Decomposition of node A52

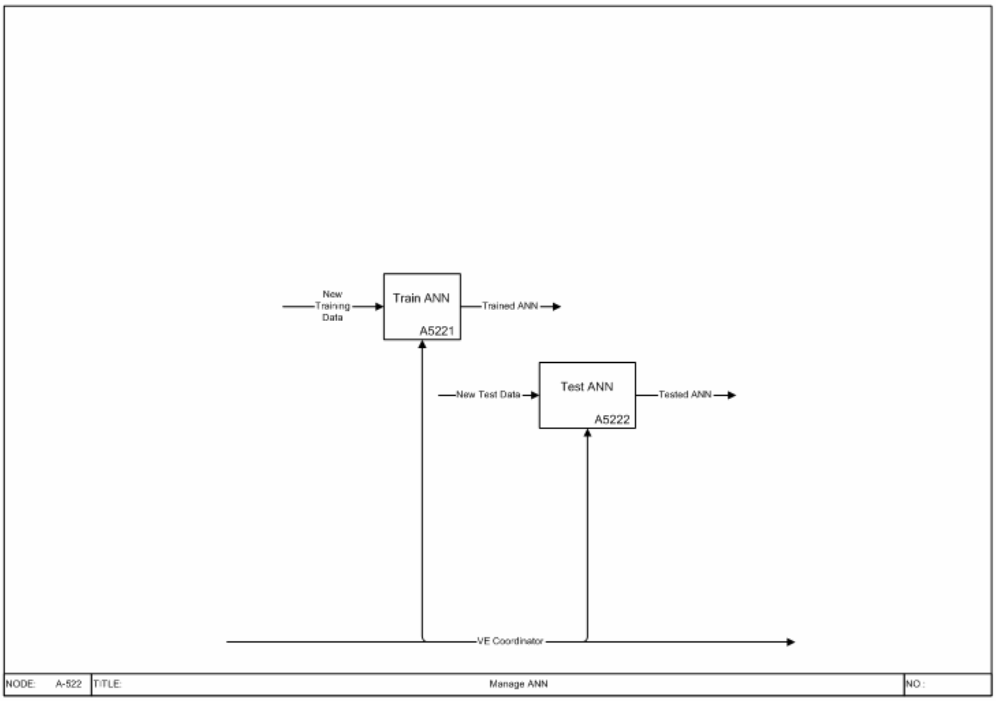


Figure A13 Decomposition of node A522

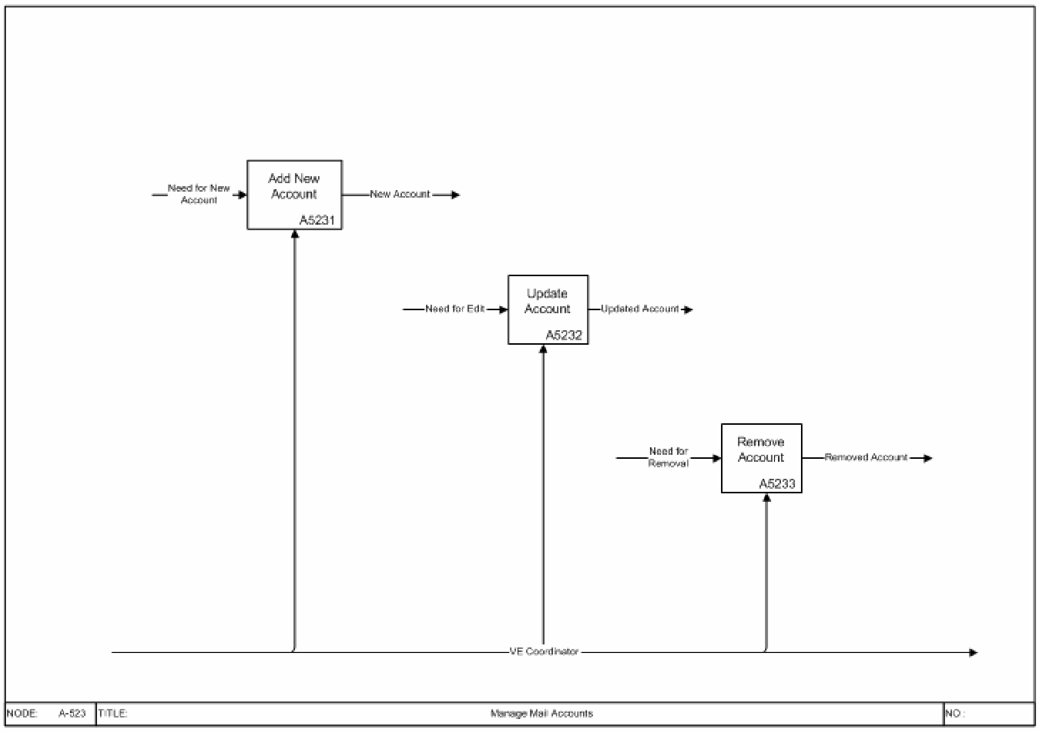


Figure A14 Decomposition of node A523

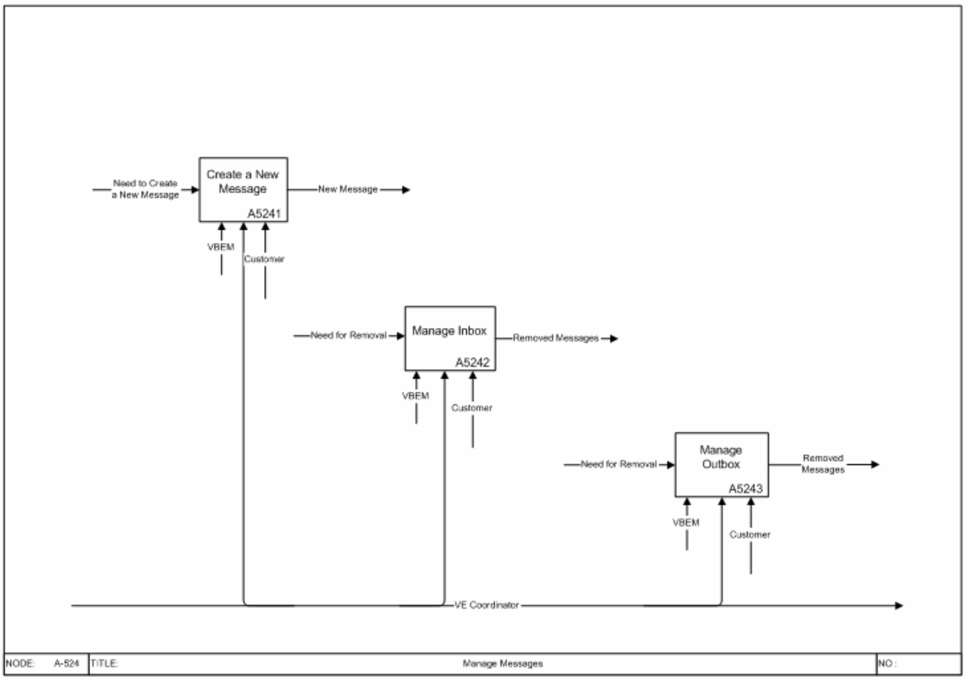


Figure A15 Decomposition of node A524

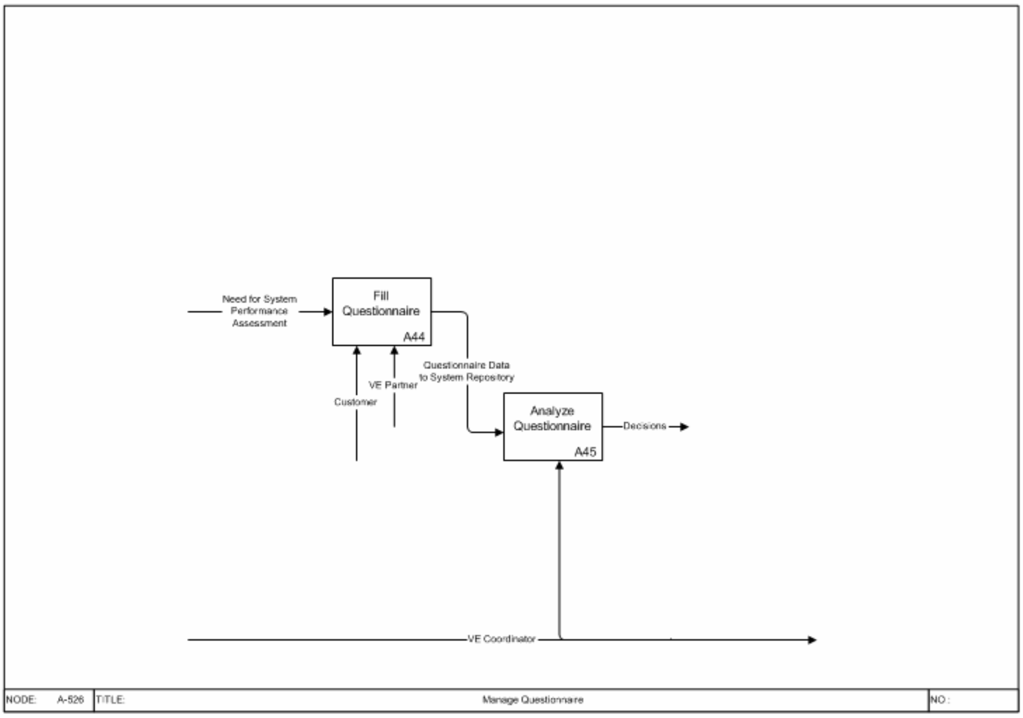


Figure A16 Decomposition of node A526

A.2 Use Cases

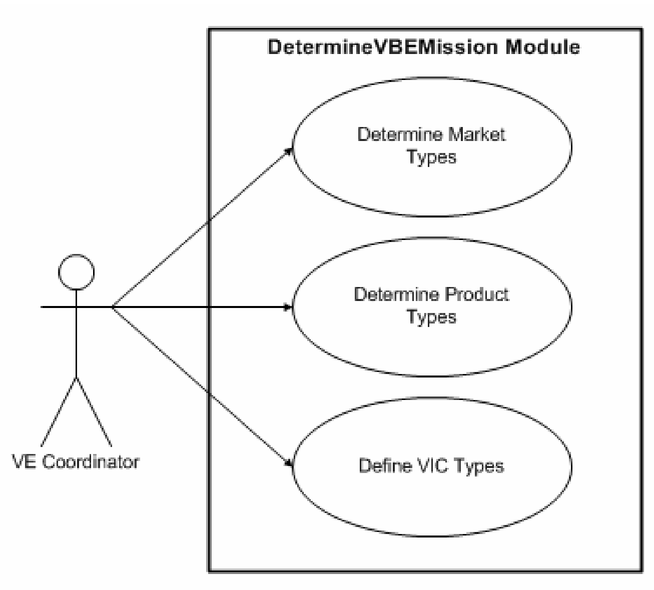


Figure A17 Use case diagram of “Determine VBE Mission” module

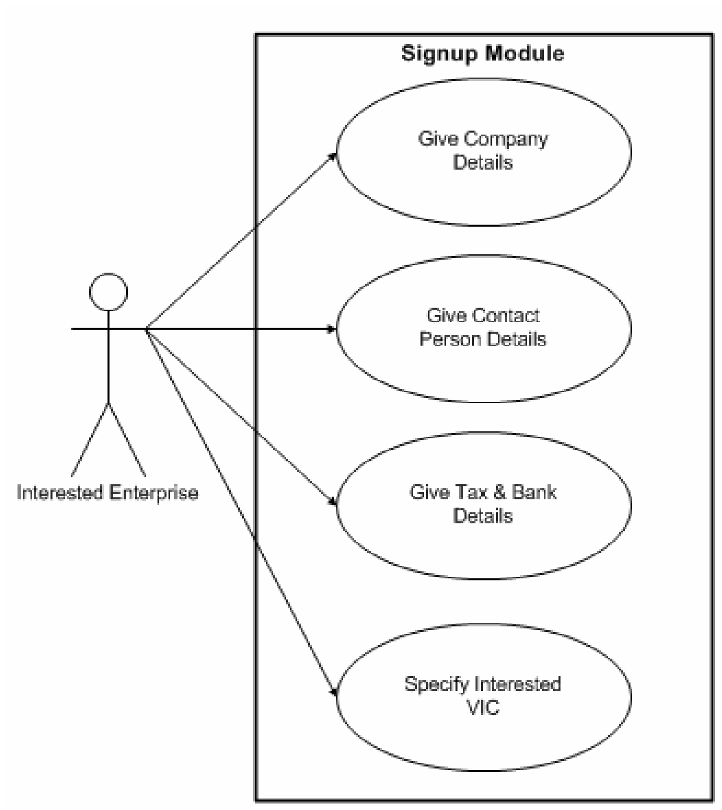


Figure A18 Use case diagram of "IE Signup" module

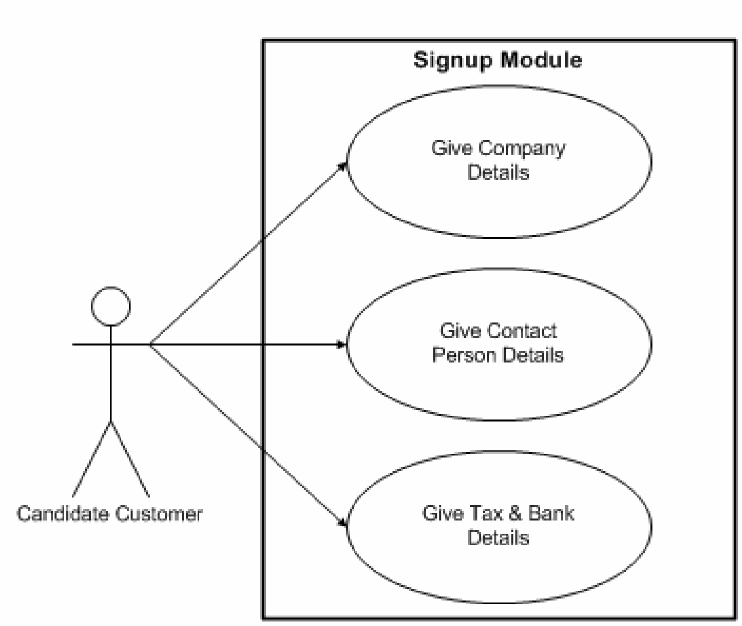


Figure A19 Use case diagram of "CC Signup" module

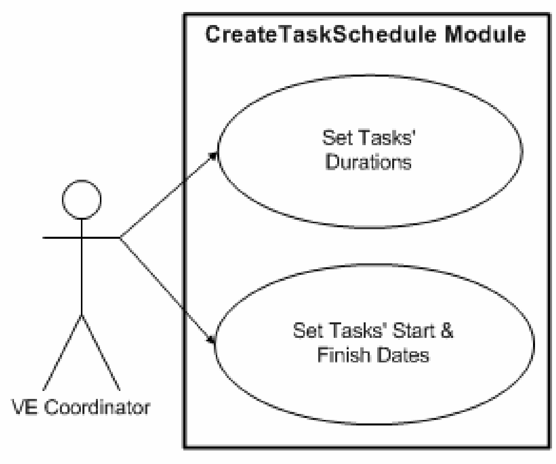


Figure A20 Use case diagram of “Create Task Schedule” module

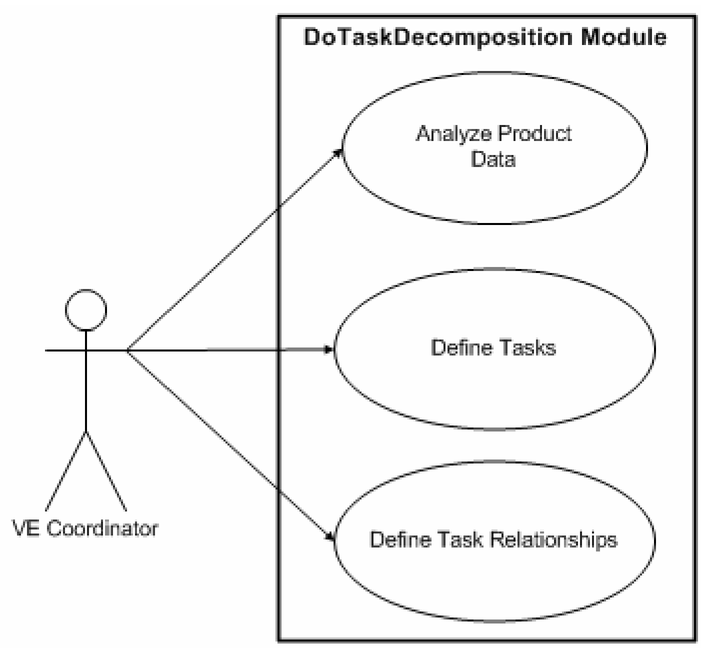


Figure A21 Use case diagram of “Task Decomposition” module

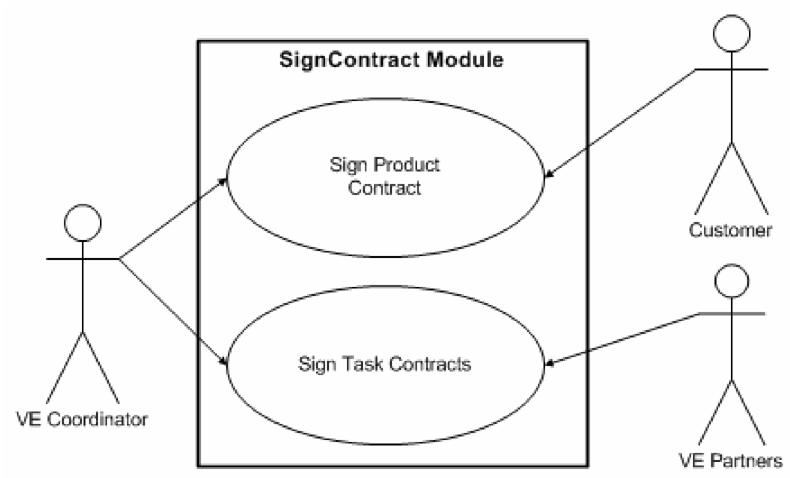


Figure A22 Use case diagram of "Sign Contract" module

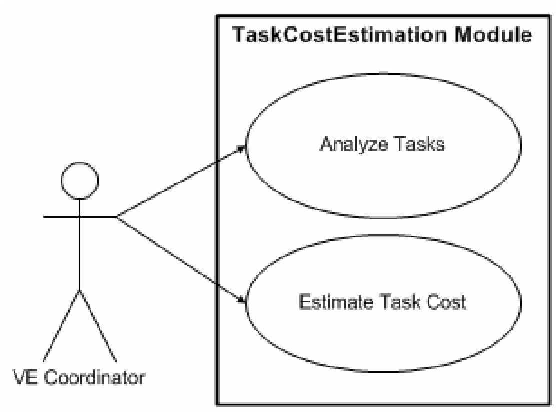


Figure A23 Use case diagram of "Cost Estimation" module

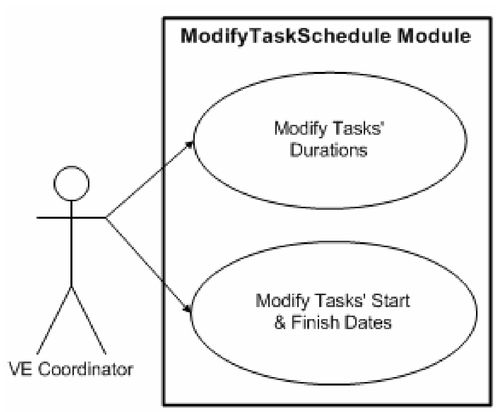


Figure A24 Use case diagram of "Modify Task Schedule" module

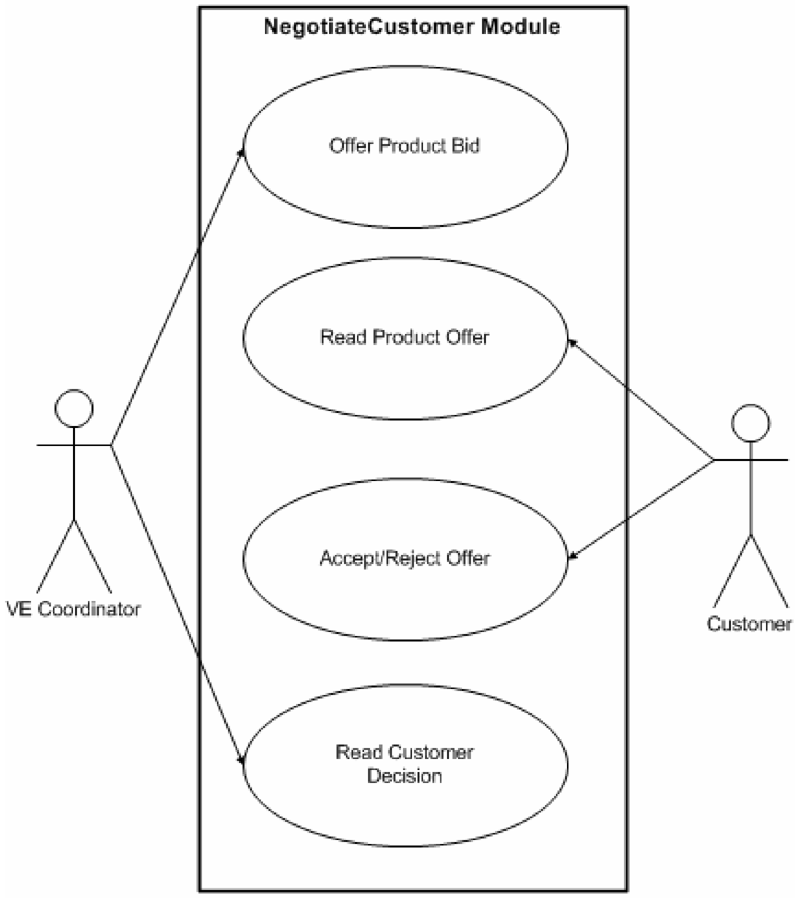


Figure A25 Use case diagram of “Negotiate with Customer” module

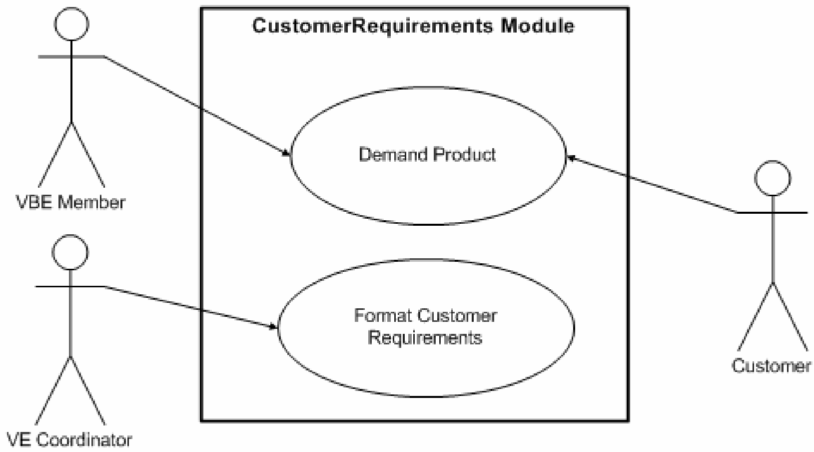


Figure A26 Use case diagram of “Identify Requirements” module

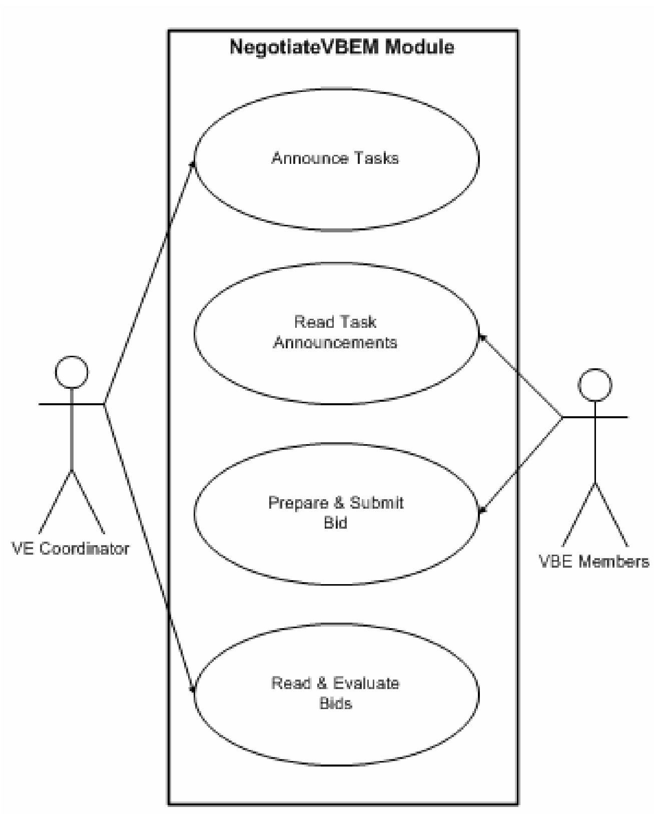


Figure A27 Use case diagram of "Negotiate with Member" module

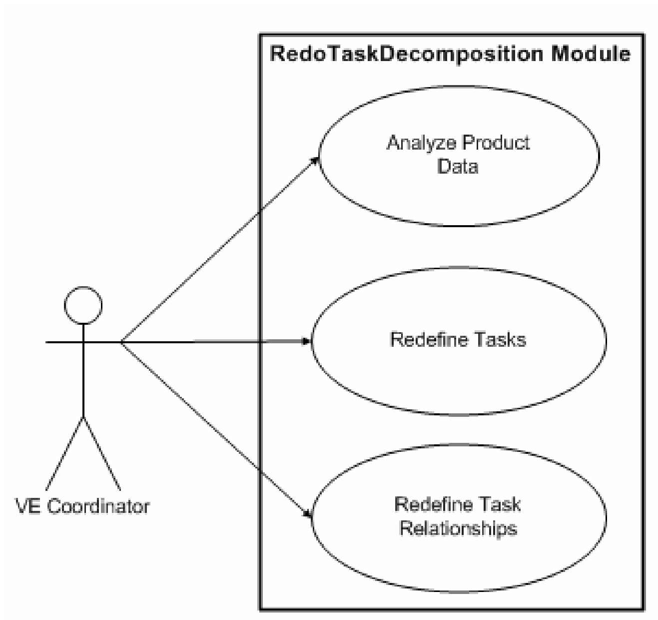


Figure A28 Use case diagram of "Redo Task Decomposition" module

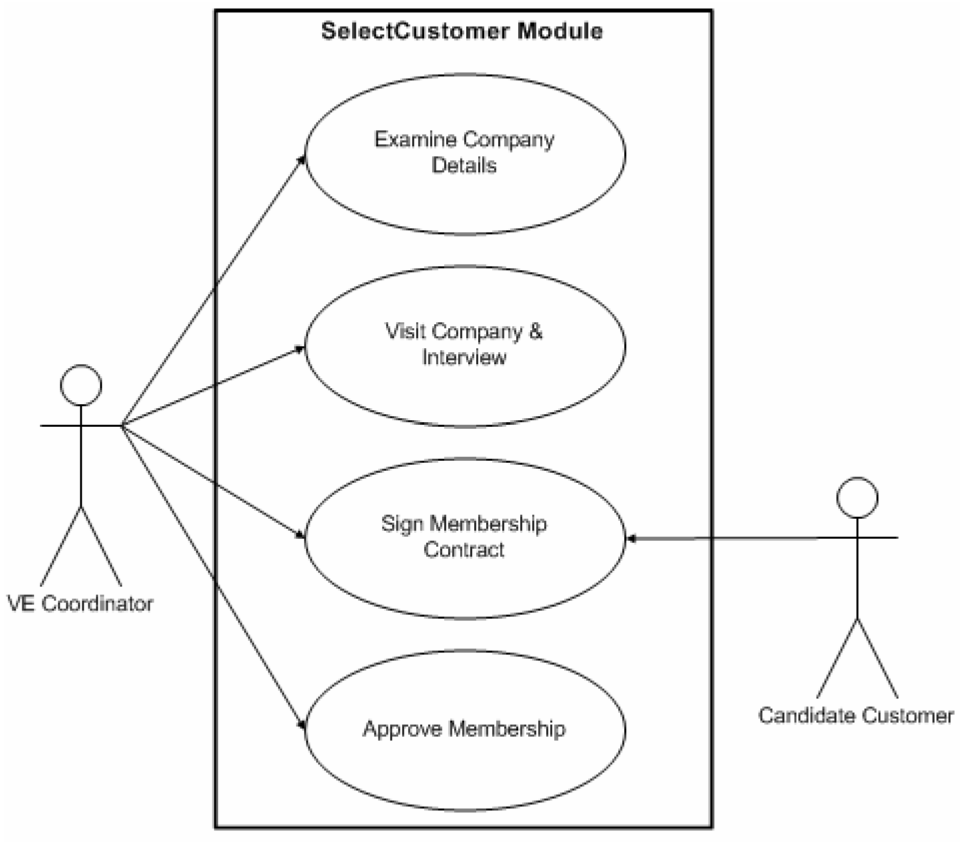


Figure A29 Use case diagram of "Select Customer" module

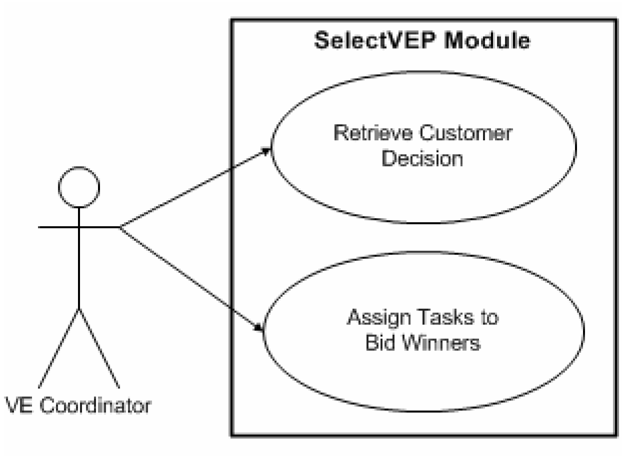


Figure A30 Use case diagram of "Select Partner" module

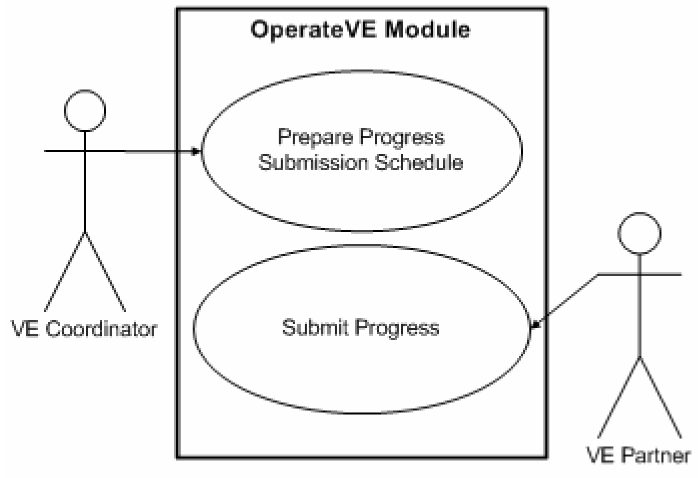


Figure A31 Use case diagram of "Operate VE" module

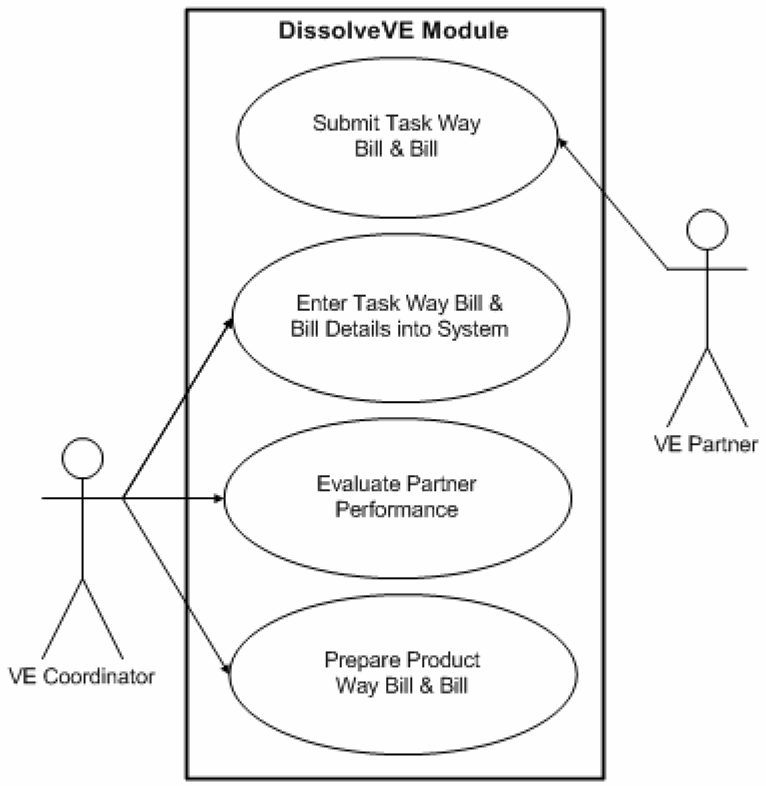


Figure A32 Use case diagram of "Dissolve VE" module

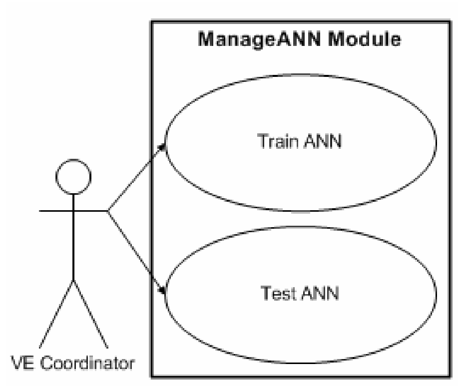


Figure A33 Use case diagram of "Manage ANN" module

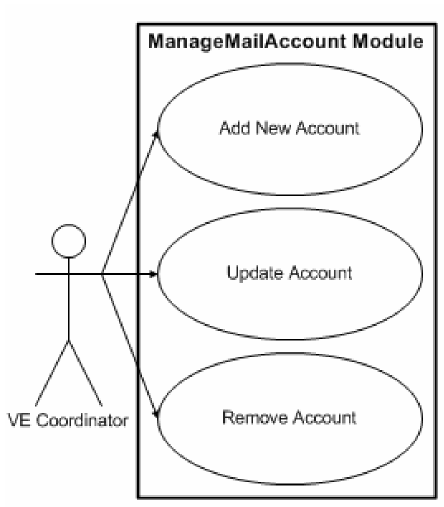


Figure A34 Use case diagram of "Manage Mail Account" module

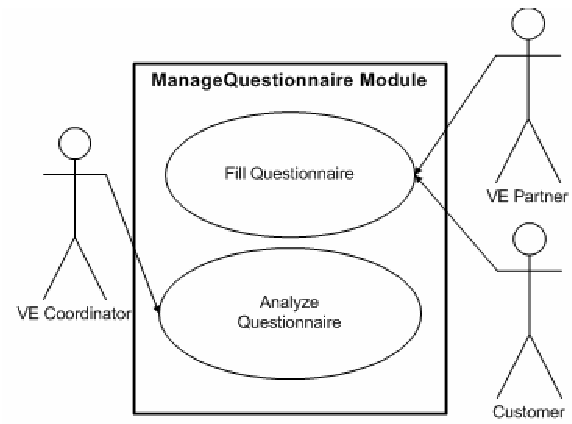


Figure A35 Use case diagram of "Manage Questionnaire" module

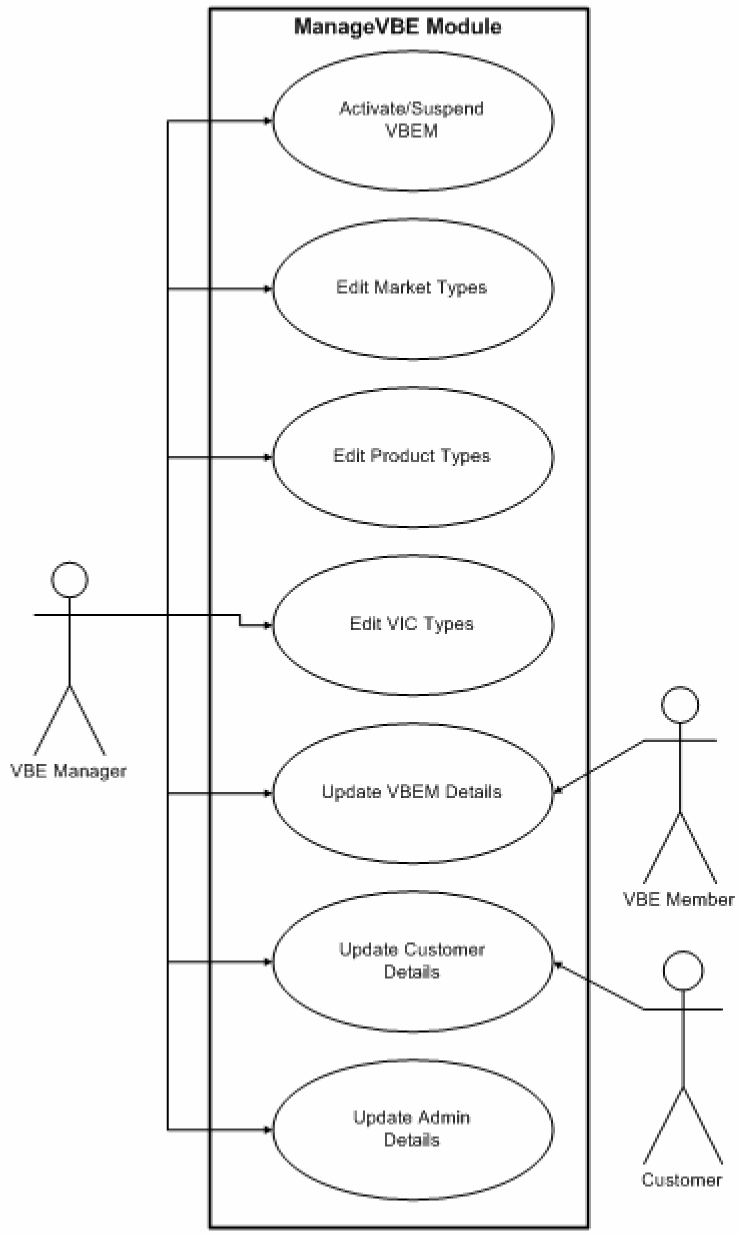


Figure A36 Use case diagram of “Manage VBE” module

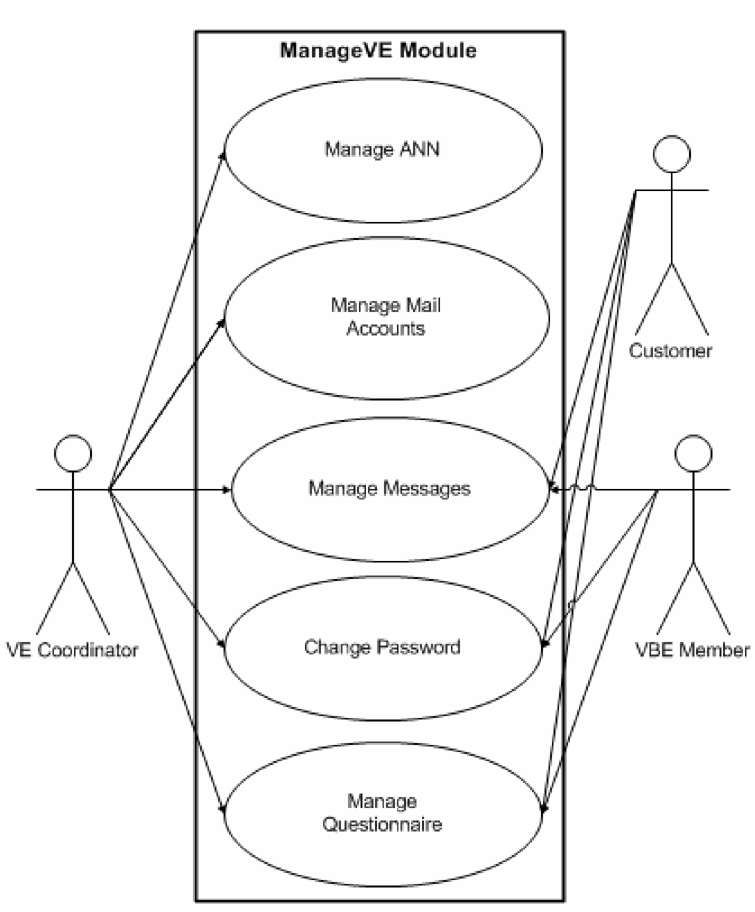


Figure A37 Use case diagram of "Manage VE" module

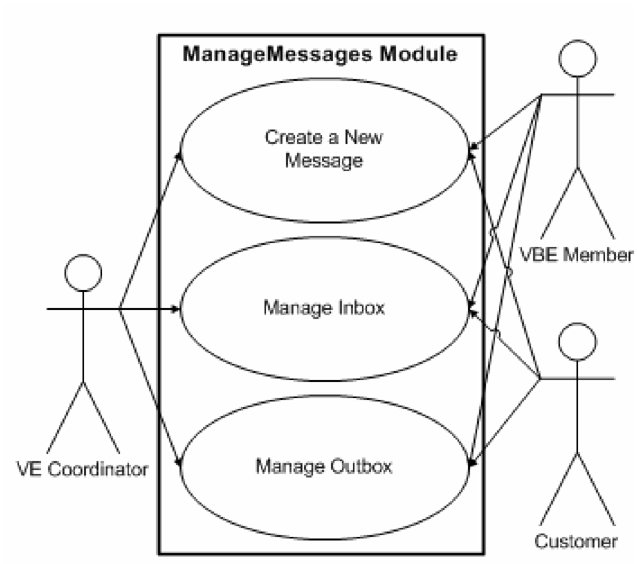


Figure A38 Use case diagram of "Manage Messages" module

A.3 Activity Diagrams

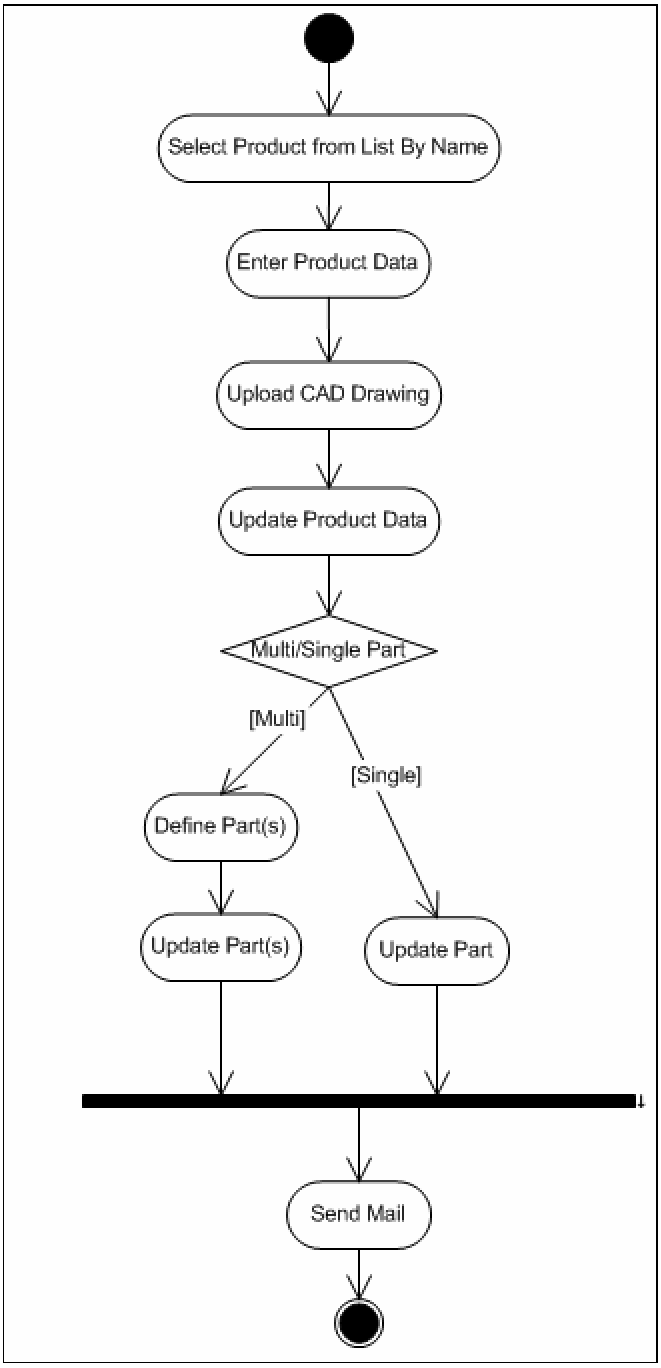


Figure A39 Activity diagram for “Product Demand”

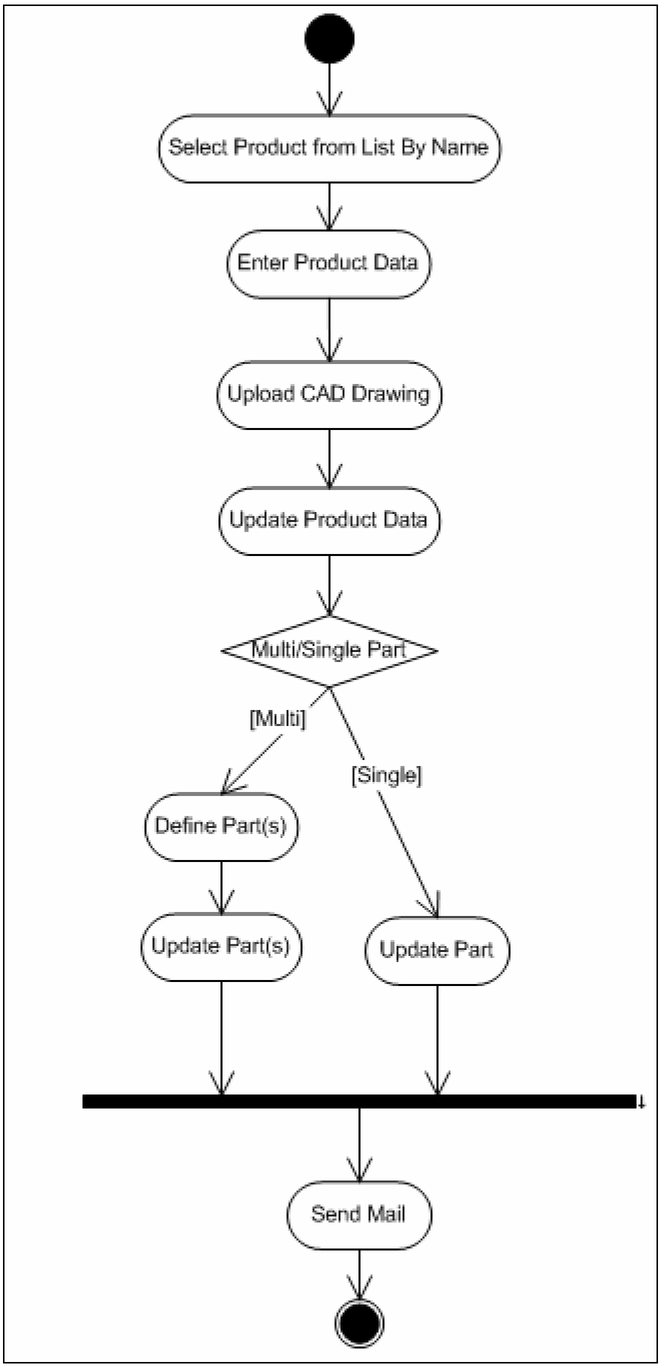


Figure A40 Activity diagram for "Determination of VBE Mission"

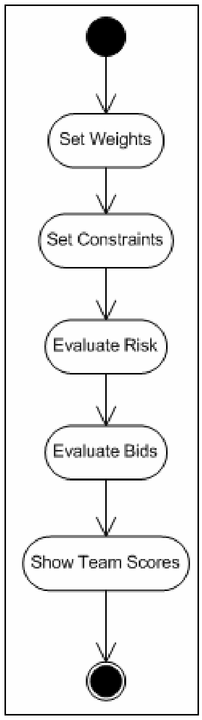


Figure A41 Activity diagram for “Bid Evaluation”

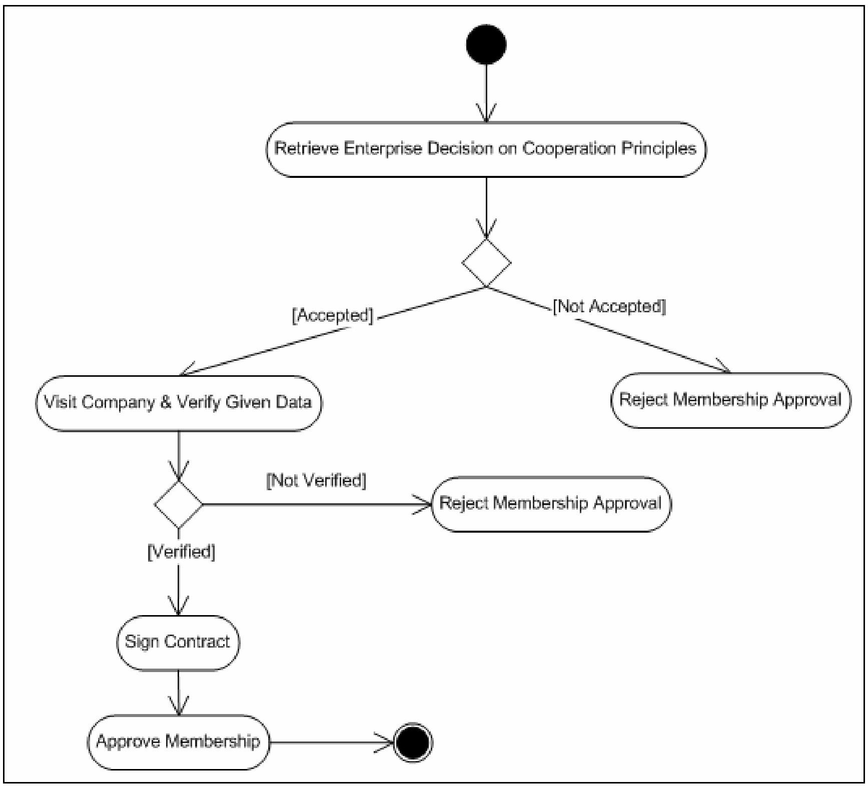


Figure A42 Activity diagram for “Membership Evaluation”

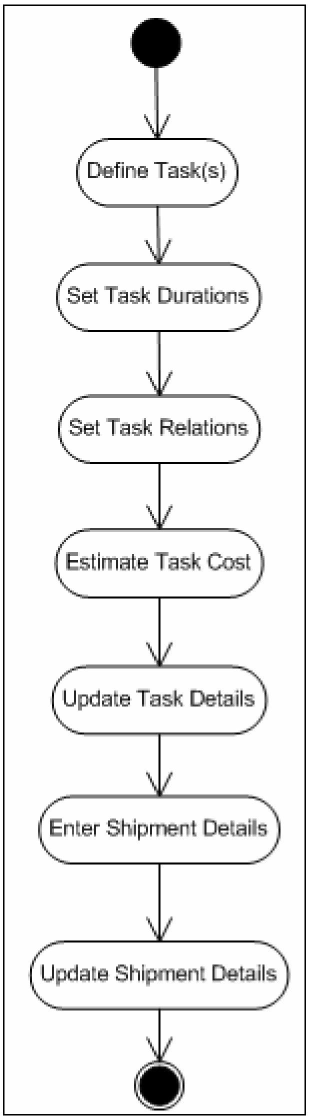


Figure A43 Activity diagram for "Task Decomposition"

A.4 Sequence Diagrams

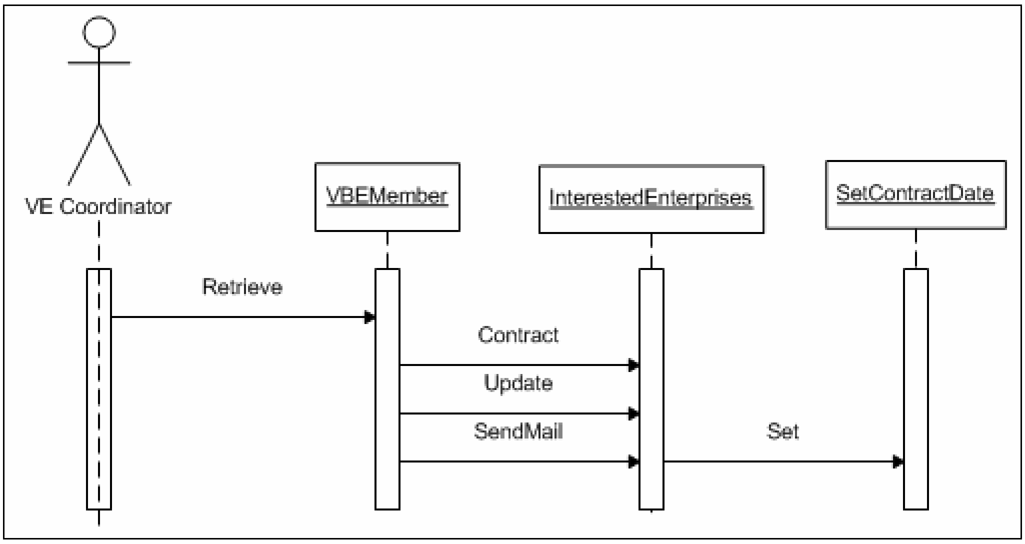


Figure A44 Sequence diagram for “VBEM Selection”

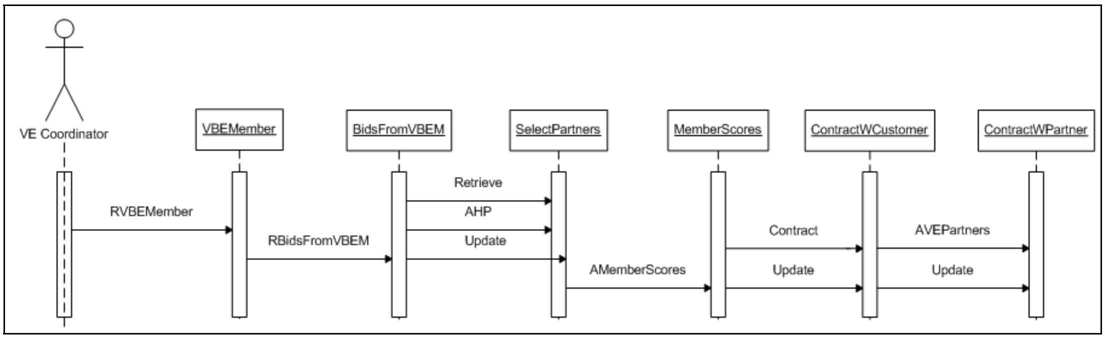


Figure A45 Sequence diagram for “Partner Selection”

APPENDIX B

USERS MANUAL

B.1 Installation

The work includes a complete web site and a database in SQL Server 2000. For an installation please follow the indicated steps:

- Install the web site “VE” on the IIS of the web server by copying all pages in “\WebSite” directory on the CD supplied.
- Install the database “\Database” on the SQL Server 2000 of the database server by selecting restore database command from the SQL Server menu.
- Copy & paste the contract templates from the “\Ftproot” directory of CD into the following path;“C:\Inetpub\ftproot”.
- Install the following programs in the web server; Adobe Acrobat, Microsoft Word, Microsoft Excel and Microsoft Project 2003.

Note that the “VE” system requires administrator assistance in installation of numerous distributed program components.

B.2 Help

The web site is published on http://imtrg-3.imtrg.me.metu.edu.tr/VE. The web site can be viewed both in Turkish and English. One can reach a detailed explanation about the use of the VE from the help service of the SME based Virtual Enterprise System (SVE) (by clicking on the help button in the below toolbar) . This online help contains information on use of the system and frequently asked questions. Please note that due to there are three different user roles in the system, help pages are differing according to the user role. The complete web site is also on the "\website\" directory located on the CD. To publish these pages on an IIS server create a new folder on IIS Web "VE" directory for example "C:\InetPub\wwwRoot\VE" and simply copy all files there. Below figures show the some of the published online help pages of the web site.

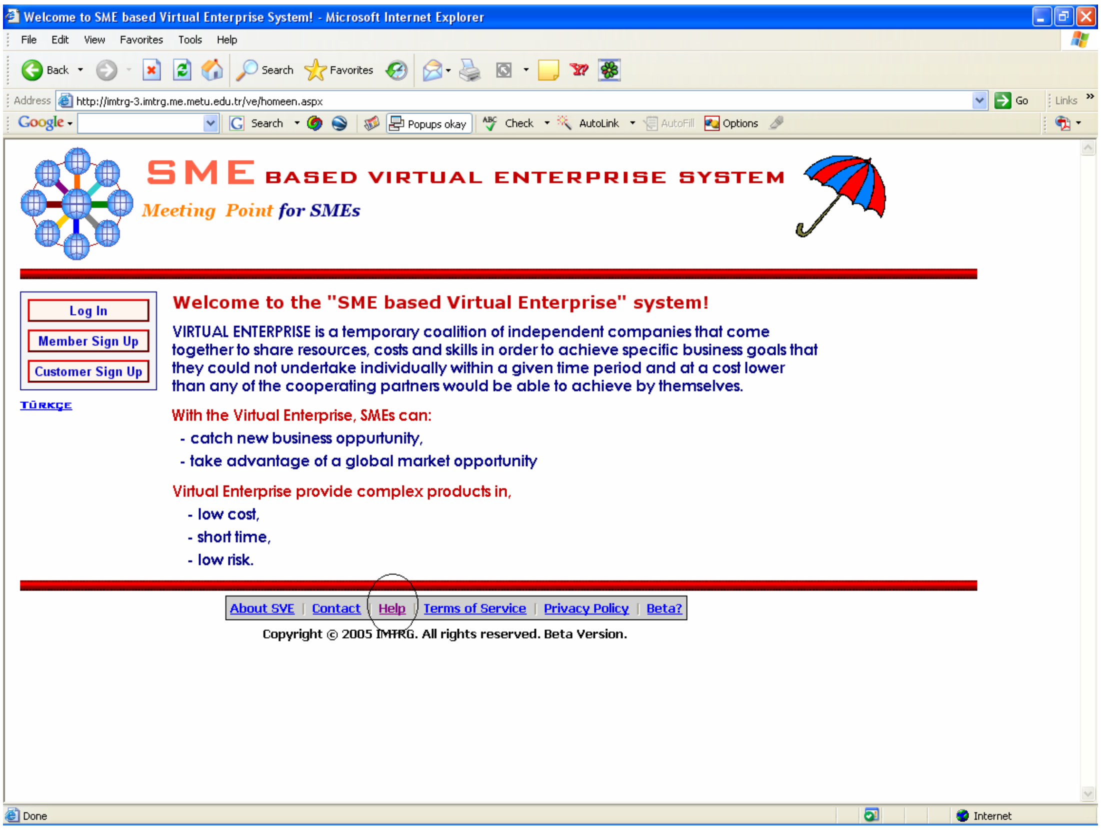


Figure B1 Help button

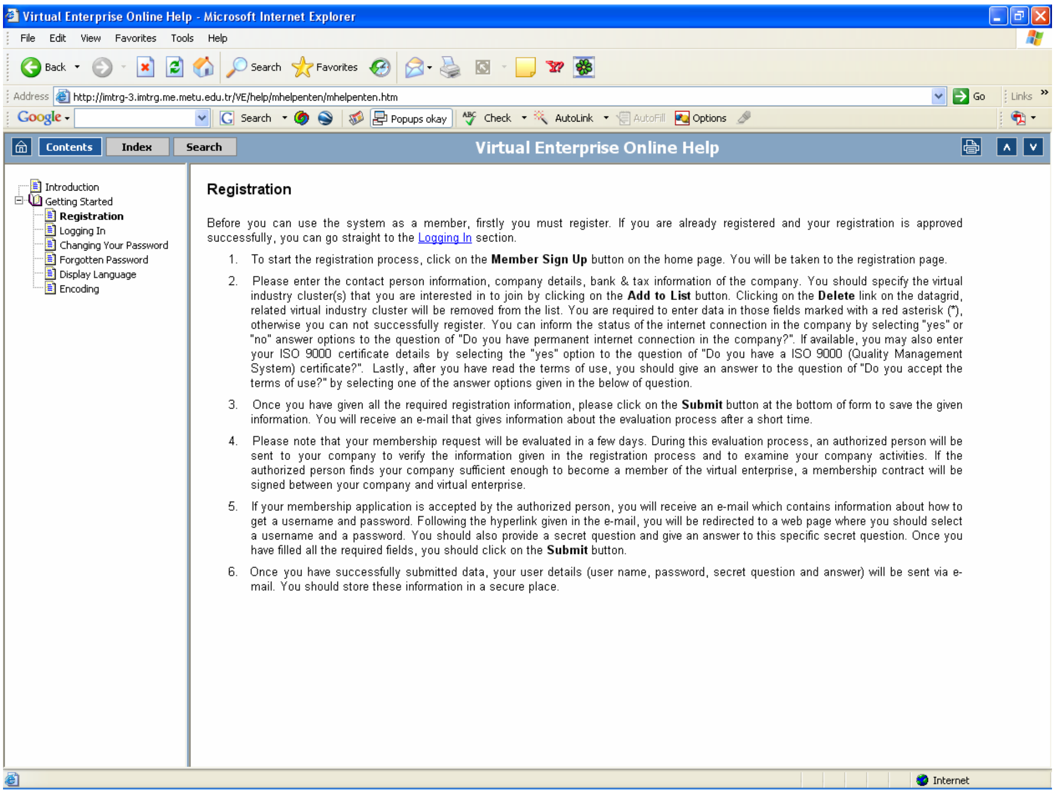


Figure B2 Help about the registration

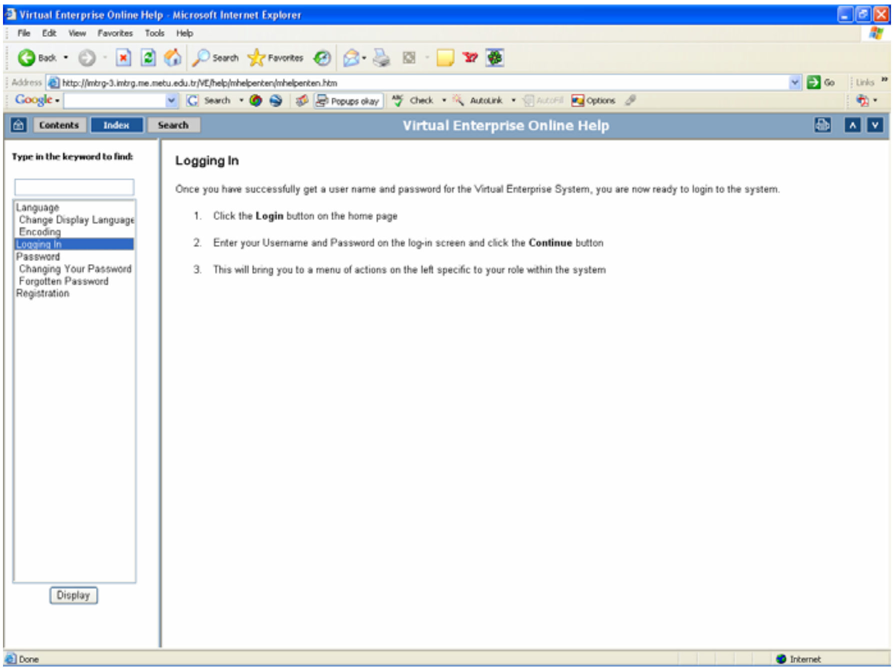


Figure B3 Help about log in

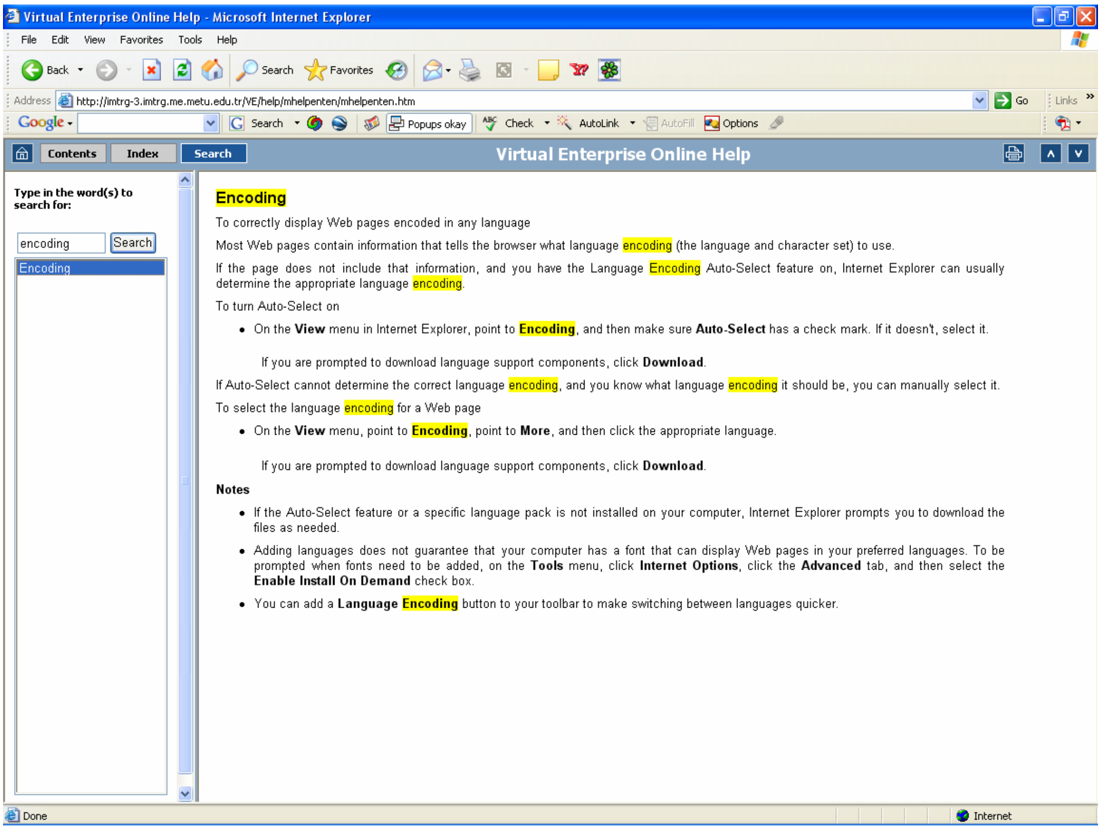


Figure B4 Help about encoding

APPENDIX C

SAMPLE CODE

The following code is taken from the MarketType objects Submit method. It shows the basic use of ADD, REMOVE, MODIFY functions and use of Try Catch block for error handling:

```
Private Sub Submit()
    Try
        Dim cnn As New SqlClient.SqlConnection(Constants.SQLConnectionString)
        cnn.Open()
        '~~~~~~~~Add~~~~~~~~~~
        If Session("button") = "Add" Then
            Dim cmdInsert As New SqlClient.SqlCommand("spAMarketType")
            cmdInsert.CommandType = CommandType.StoredProcedure
            cmdInsert.Connection = cnn
            With cmdInsert.Parameters
                .Add("@prmMarket_Type_ID", SqlDbType.Int, 4,
"Market_Type_ID").Value = txtMarketID.Text
            End With
            cmdInsert.ExecuteNonQuery()
            Dim cmdUpdate As New SqlClient.SqlCommand("spMMarketType")
            cmdUpdate.CommandType = CommandType.StoredProcedure
            cmdUpdate.Connection = cnn
            With cmdUpdate.Parameters
                .Add("@prmMarket_Type_ID", SqlDbType.Int, 4,
"Market_Type_ID").Value = txtMarketID.Text
```

```
                .Add("@prmMarket_Type_Name", SqlDbType.NVarChar, 50,
"Market_Type_Name").Value = txtMarketName.Text
                .Add("@prmMarket_Type_NameT", SqlDbType.NVarChar, 50,
"Market_Type_NameT").Value = txtMarketNameT.Text
            End With
            cmdUpdate.ExecuteNonQuery()
            cnn.Close()
            cboMarket.Items.Clear()
            SubmitFillComboBox(strM)
            btnAdd.Enabled = True
            btnRemove.Enabled = True
            btnModify.Enabled = True
            btnSubmit.Enabled = False
            btnCancel.Enabled = False
            txtMarketName.ReadOnly = True
        ElseIf Session("button") = "Delete" Then
            Dim cmdRemove As SqlClient.SqlCommand
            Dim strsql As String
            strsql = "spDMarketType " & txtMarketID.Text
            cmdRemove = New SqlClient.SqlCommand(strsql, cnn)
            cmdRemove.CommandType = CommandType.Text
            cmdRemove.ExecuteNonQuery()
            cnn.Close()
            cboMarket.Items.Clear()
            FillComboBox()
            btnAdd.Enabled = True
            btnRemove.Enabled = True
            btnModify.Enabled = True
            btnSubmit.Enabled = False
            btnCancel.Enabled = False
            txtMarketName.ReadOnly = True
        ElseIf Session("button") = "Modify" Then
```

```
            Dim cmdModify As New SqlClient.SqlCommand("spMMarketType")
            cmdModify.CommandType = CommandType.StoredProcedure
            cmdModify.Connection = cnn
            With cmdModify.Parameters
              .Add("@prmMarket_Type_ID", SqlDbType.Int, 4,
"Market_Type_ID").Value = txtMarketID.Text
              .Add("@prmMarket_Type_Name", SqlDbType.NVarChar, 50,
"Market_Type_Name").Value = txtMarketName.Text
              .Add("@prmMarket_Type_NameT", SqlDbType.NVarChar, 50,
"Market_Type_NameT").Value = txtMarketNameT.Text
            End With
            cmdModify.ExecuteNonQuery()
            cnn.Close()
            cboMarket.Items.Clear()
            SubmitFillComboBox(strM)
            btnAdd.Enabled = True
            btnRemove.Enabled = True
            btnModify.Enabled = True
            btnSubmit.Enabled = False
            btnCancel.Enabled = False
            txtMarketName.ReadOnly = True
          End If
        Catch ex As Exception
          lblError.Text = ex.ToString
        End Try
      End Sub
```

APPENDIX D

GLOSSARY

Active Server Pages	An open application environment in which HTML pages, server-side scripts, and ActiveX components are combined to create web based applications.
Business Process (BP)	A procedure where documents, information or tasks are passed between entities of the workflow according to defined sets of rules to achieve, or contribute to, an overall business goal.
Dynamic VE	Is a dynamic constellation of different administrative domains that cooperate in order to execute and manage share business processes. The form and the relationships among the partners are built dynamically and after negotiation.
Electronic Contract	Is the electronic representation of the agreement reached between two administrative domains for the sharing of a particular business process. It regulates the terms and conditions of the partnership.
IDEF	The methodologies that have been developed to analyze manufacturing systems.
Life-Cycle Model	The finite steps a system may go through over its entire life history. The different life cycle phases define types

	of activities which are pertinent during the life cycle of the entity, ISO/DIS 15704 (ISO)
Negotiation	Is a process by which a joint decision is made by two or more parties. The parties first verbalize contradictory demands and then move towards agreement by a process of concession making or search for new alternatives
Static VE	Is a static constellation of different administrative domains that cooperate in order to execute and manage pre-defined and statically specified business processes. The form and the relationships among the partners are built statically and before the provision of the process to the customer.
Task	A well-defined business activity which can not be further subdivided between a numbers of players.
VE Breeding Environment	Is the set of matchmaking services for the dynamic selection of partners. The matchmaking services consist of the service type management, service offer management, and service retrieval management.
VE Candidate Partner	Is the administrative domain that registers its local business processes to the virtual marketplace and is willing to establish dynamic relationships with other domains.
VE Coordinator	Is the administrative domain that represents the VE to the external world and provides the shared business processes to the different customers.

VE Partner	Is the domain that has been engaged itself into a business relationship with another domain through negotiation. The VE partner offers for a very short period of time a specific business process with specific terms and conditions to another domain. terms and conditions to another domain.
Virtual Enterprises (VE)	A network of different administrative business domains that cooperate by sharing business processes and resources to provide a value-added service to the customer.

CURRICULUM VITAE

Burak SARI
Middle East Technical University, Department of Mechanical Engineering,
B-162, 06531, ODTU/Ankara/TURKEY
90 312 210 5286 (office), 90 532 722 2998 (mobile)
90 312 210 2536 (fax)
burak@me.metu.edu.tr, sariburak@hotmail.com

PERSONAL INFORMATION

Sex: Male
Marital Status: Single
Nationality: Turkish
Date of Birth: 15.11.1977
Place of Birth: Mersin/TURKEY
Driving License: Class B

SUMMARY OF SKILLS

- Skilled research engineer with M.Sc. degree in mechanical engineering and solid background in information technologies.
- Managing and participating projects from conception to completion with high motivation and ambition, meeting deadline requirements.
- Skilled at learning new concepts quickly, communicating (oral and written) ideas clearly and effectively, analytical thinking in problem solving and ability to work as a team member or individually.
- An individual whose personal philosophy and values have enabled him to succeed, inspire and lead others.

EDUCATION

- **Master of Science** in Department of Mechanical Engineering, Middle East Technical University, Ankara, TURKEY, 1999-2001, (CGPA: 3.64/4.00), Thesis title: Development of a job shop planning system: optimization of machining operations using Windows DNA architecture
- **Bachelor of Science** in Department of Mechanical Engineering, Istanbul University, Istanbul, TURKEY, 1995-1999, (CGPA: 3.35/4.00), (Ranked 3rd over 120)
- **High School** (CGPA: 4.01/5.00), Tarsus American College, Mersin, Turkey, 6 years high school including one year preparatory class, Completely in English

EXPERIENCE

Research Assistant in Department of Mechanical Engineering, Middle East Technical University, Ankara, TURKEY, 1999-2006

PUBLICATIONS

- Sarı B., Kılıç S.E., and Şen D.T., "Formation of Dynamic Virtual Enterprises and Enterprise Networks", International Journal of Advanced Manufacturing Technology, 2006, **ACCEPTED**
- Sarı B., Amaitik S., and Kılıç S. E., "A Neural Network Model for the Assessment of Partners' Performance in Virtual Enterprises", International Journal of Advanced Manufacturing Technology, 2006, DOI 10.1007/s00170-006-0642-z.
- Sarı B., Kılıç S. E., and Şen D. T, "Dağıtık İmalat Teknolojilerinde KOBİ'lere Yönelik Sanal Fabrika Sisteminin Geliştirilmesi", Machinery MakinaTek Aylık İmalat ve Teknoloji Kültürü Dergisi, pp. 108-112, November 2005 – pp. 108-111, December 2005.
- Sarı B., Kılıç S. E., and Şen D. T, "Dağıtık İmalat Teknolojilerinde KOBİ'lere Yönelik Sanal Fabrika Sisteminin Geliştirilmesi", TMMOB Makina Mühendisleri Odası Konya Şubesi III. Makina Tasarım ve İmalat Teknolojileri Kongresi, Konya, Turkey, pp. 65-75, September 16-17, 2005.

- Sarı B., Kılıç S. E., "Development of an SME-based Virtual Enterprise", 11th International Conference on Machine Design and Production, 11th International Machine Design and Production Conference, Antalya, Turkey, pp. 79-93, October 13-15, 2004.
- Sarı B., Kılıç S. E., and Şen D. T, "Talaşlı İmalat için Web Tabanlı Optimizasyon Sistemi", TMMOB Makina Mühendisleri Odası Konya Şubesi II. Makina Tasarım ve İmalat Teknolojileri Kongresi, Konya, Turkey, pp. 3-12, September 26-27 2003.
- Sarı B., Kılıç S. E., and Şen D. T, "Web Based Optimization System for Machining Operations", RADMI 2003 (3rd International Research and Development in Mechanical Industry), Herceg-Novi, Serbia and Montenegro, pp. 1557-1562, September 19-23, 2003.
- Sarı B., Amaitik S. and Kılıç S. E., "A Neural Network Model for Selection of Machining Parameters in CAPP", WESIC 2003 (4th Workshop on European Scientific and Industrial Collaboration), Miskolc, Hungary, pp. 355-362, 2003.
- Başıbüyük Y., Sarı B., Kılıç S. E., and Anlağan Ö., Agent-Based Shop Floor Control System using Windows DNA Technology", UMTiK 2002 The 10th International Conference on Machine Design and Production, pp. 225-236, September 4-6, 2002.
- Sarı B., Kılıç S. E., Anlağan Ö., and Şen D. T., "Web-Based Optimization Systems for Machining Operations", UMTiK 2002 The 10th International Conference on Machine Design and Production, pp. 259-278, September 4-6, 2002.
- Sarı B., Akbaş M., Anlağan Ö. and Kılıç S. E., "G Code Generation for Turning", International Conference on Millennium Dawn in Training and Continuing Education, University of Bahrain, Bahrain, pp. 123-137, April 24-26, 2001.
- Sarı B., Cangar T., Anlağan Ö. and Kılıç S. E., "Web Based CIM Laboratory Experience in ME Curriculum: Part Design, NC-Code Generation and Work Order Dispatching via Internet", Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition, American Society for Engineering Education, pp. 11367-11388, 2001.

- Sarı B., Anlağan Ö. and Kılıç S. E., “An Optimization System For Milling and Turning Operations Based On Windows DNA Architecture”, XII Workshop on Supervising and Diagnostics of Machining Systems, Karpacz, Poland, pp. 24-34, March 18-23, 2001.

PROJECTS

- Methodology Development for SME based Virtual Enterprises; Supported by the Scientific and Technical Research Council of Turkey (TUBITAK), 2004-2006
- Development of a Manufacturing Execution System for SME's; Supported by the Scientific and Technical Research Council of Turkey (TUBITAK), 2002-2004
- Development of an Agent-based Virtual Manufacturing System using Microsoft Windows DNA Architecture; Supported by Turkish State Planning Agency (DPT), 2001-2003
- Development of an Educational Test Bed for the Implementation of Distributed Internet Applications in Manufacturing Execution System; Supported by the Scientific and Technical Research Council of Turkey (TUBITAK), 2001-2003
- Development of a Shop Floor Control System using CORBA for Agile Manufacturing; Supported by Turkish State Planning Agency (DPT), 1999-2003