DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A FRAMEWORK BASED ON QUALITY FUNCTION DEPLOYMENT FOR REQUIREMENTS ANALYSIS IN ENTERPRISE MODELLING

by
Güzin ÖZDAĞOĞLU

October, 2009
İZMİR

A FRAMEWORK BASED ON QUALITY FUNCTION DEPLOYMENT FOR REQUIREMENTS ANALYSIS IN ENTERPRISE MODELLING

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Industrial Engineering Program

by
Güzin ÖZDAĞOĞLU

October, 2009
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A FRAMEWORK BASED ON QUALITY FUNCTION DEPLOYMENT FOR REQUIREMENTS ANALYSIS IN ENTERPRISE MODELLING"** completed by **GÜZİN ÖZDAĞOĞLU** under supervision of **ASSOC.PROF.DR. LATİF SALUM** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for Doctor of Philosophy.

ASSOC.PROF.DR.LATİF SALUM

Supervisor

ASST.PROF.DR.ŞEYDA TOPALOĞLU

Thesis Committee Member

ASST.PROF.DR.ADİL ALPKOÇAK

Thesis Committee Member

PROF.DR. GÜLSER KÖKSAL

Examining Committee Member

ASST.PROF.DR.MEHMET ÇAKMAKÇI

Examining Committee Member

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to thank and emphasize my appreciation to my advisor Dr. Latif Salum for his great and invaluable advice and support that guided me to form my considerations in the study and complete this dissertation. I would like to thank to thesis committee members Dr. Şeyda Topaloğlu and Dr. Adil Alpkoçak for their distinguished and helpful suggestions and support. I also would like to thank my family, especially to my husband and my colleague Dr. Aşkın Özdağoğlu for his support, help and encouragement.

Finally, I would like to acknowledge Dr. Sabri Erdem and other colleagues in DEU Faculty of Business for their support during my Ph.D. study.

A FRAMEWORK BASED ON QUALITY FUNCTION DEPLOYMENT FOR REQUIREMENTS ANALYSIS IN ENTERPRISE MODELLING

ABSTRACT

Competitiveness and globalization force enterprises to quickly adapt to changing conditions of markets. Enterprises employ some modelling methodologies to organize their strategic knowledge to cope with this change, which results in an enterprise model. Requirements discovery and analysis is the most important phase in creating the enterprise model because any mistake in the requirements discovery deteriorates the validity of the model, resulting in user dissatisfaction. Quality Function Deployment (QFD) is a well-known and integrated approach used in converting the requirements of users into final product specifications. This thesis modifies QFD for enterprise modelling, and proposes Enterprise-QFD, which provides a common platform that can be integrated with any methodology for discovering and analyzing enterprise requirements. The study synthesizes enterprise modelling, requirements analysis and modelling, and QFD concepts and proposes an approach based on Modern QFD to analyze the requirements of an enterprise from the long term goals to the functional, informational, organizational, and resource characteristics. The modified QFD tables involve some required columns added and unnecessary ones deleted based on enterprise modelling. A novel matrix content and sequence is also proposed. In the scope of the study, Enterprise-QFD is applied to a small business company processing steel products with real evaluations and the findings to show the usability of the method. After the requirements are analyzed and modelled by Enterprise-QFD, the findings are transferred to the requirements model of CIMOSA, a complicated enterprise reference architecture. The results show that Enterprise-QFD generates the infrastructure for further modelling of enterprise architectures concerning both functional characteristics of enterprise and needs of stakeholders.

Keywords: Enterprise Modelling, Enterprise Architecture, Requirement Analysis And Modelling, Quality Function Deployment, CIMOSA.

KURUM MODELLEMESİNDE GEREKSİNİMLERİN ANALİZİ İÇİN KALİTE FONKSİYON GÖÇERİMİ TABANLI BİR ÇERÇEVE YAPI

ÖZ

Rekabet ve küreselleşme kurumları Pazar koşullarındaki değişimlere hızlı uyum sağlamaya zorlamaktadır. Kurumlar bu hızlı değişimlerle başa çıkabilmek ve stratejik bilgi ve deneyimlerini organize etmek amacıyla kurum modeli adı verilen metodolojiler uygulamaktadır. Bir kurum modelinin oluşturulmasında en önemli aşama kurum ihtiyaçlarının keşfi ve analizidir. Çünkü, ihtiyaçların keşfedilmesinde yapılacak herhangi bir hata, yaratılan modelin geçerliliğini zedeleyecek ve kullanıcılar açısından memnuniyetsizlikle sonuçlanacaktır. Kalite Fonksiyon Göçerimi (KFG), bütünleşik yapısıyla kullanıcı ihtiyaçlarının son ürün özelliklerine dönüştürmesindeki başarısıyla tanınan bir ihtiyaç analizi metodolojisidir. Bu tez çalışması, kurum modeli için kullanılabilir hale getirilmesi amacıyla KFG metodolojisinin araçlarında bir takım değişimler önermektedir. Oluşturulan bir çerçeve model ışığında bu çalışma, kurum modellemesinin, ihtiyaç analizi ve modellemesinin ve KFG metholodojisinin bir sentezini sunmakta ve kurum içinde uzun dönem hedeflerinden fonksiyon, bilgi, organizasyon ve kaynak karakteristiklerine kadar analiz eden bir çerçeve model (Enterprise-QFD) önermektedir. Bu karakteristikleri oluşturmak adına, KFG'de kullanılan tablolara gerekli yeni sütunlar eklenmiş, kullanılmayanlar çıkarılmış, sayısal analizlerde kullanılan matris yapısı ve uygulama dizisi, kurum modeli için yeniden organize edilmiştir. Bu çalışma kapsamında, önerilen yaklaşımın uygulanabilirliği, küçük ölçekli bir kurumdan elde edilen veriler üzerinde gösterilmiştir. İhtiyaç analizinin tamamlanmasının ardından elde edilen bilgiler, CIMOSA referans mimarisi baz alınarak, ihtiyaç modellemesi seviyesinde, analiz sonuçları kurum modeline yansıtılmıştır. Önerilen yaklaşımın uygulama sonuçları karmaşık bir yapıya sahip olan bu kurum mimarisinin gereksinimlerini kolaylıkla karşılamaktadır.

Anahtar sözcükler: Kurum modeli, Kurum Mimarisi, İhtiyaç Analizi ve Modellemesi, Kalite Fonksiyon Göçerimi, CIMOSA.

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

Enterprises are living organisms which can be affected by environmental conditions and internal dynamics. This effect may obstruct the enterprise from improving in Market. Survival of an enterprise in its own industry depends on the well-managed system and the performance of the processes working on this system. Environmental competency conditions forces enterprises to adapt to a rapid change. Enterprises which cannot adapt to these conditions may lose its core competence. This reality emphasizes that the systems and the processes within an enterprise may frequently face to a continuous change. Enterprises should include a constant component or a mechanism that manages the changing activities. This mechanism can only be a macro model that provides working on all subsystems integrated as a unique skeleton. This concept is generally called as *enterprise integration*. Constructing a proper integration within an enterprise indicates a particular infrastructure based on a well-built model. Thus, the integration is a long run process requiring detailed observations and analysis working on the subsystems that have been established for various goals.

The major subsystem established in an enterprise is the quality management systems, e.g. ISO 9000/9001. Such systems defining all system components, goals, processes, performance criteria, data and documentation flows are naturally in interaction with the other subsystems. ISO 14001, OHSAS 18001, and SA 8000 are some other subsystems working on the enterprise system. Besides, there exist many other subsystems, e.g. ISO 16949, ISO 22000 specialized to the industry in which enterprises perform their operations. The common properties of all these subsystems are that they require intensive documentation. These standards frequently require process improvements in many points within the main system and these improvements are planned in particular programs based on specific tools and models. Lean manufacturing (value stream mapping, 5S, TPM, and etc.), product improvement models (Quality Function Deployment (QFD) and related tools), Six Sigma, decision support systems, computer integrated manufacturing, enterprise application systems, supply chain management, customer relationships management are such models and subsystems that are used to manage particular operations and activities in an enterprise. All the business standards and other programs that are performed to manage the enterprise have intersections and they should be integrated in such a jigsaw puzzle pieces so that they serve for common enterprise goals. In other words, all the subsystems should work as a unique system working for the same goals, and enterprise integration provides such a platform. Otherwise, each system may work as a single system maintaining its own operations and targets and this situation may probably cause many repetitions and waste work.

Recent applications to obtain enterprise integration are to establish a macro model and provide all subsystems working on this model as a unique body. This model is generally described as *enterprise model.* An enterprise model is a macro model that defines the framework of an enterprise and provides an infrastructure to integrate the subsystems. Therefore the main necessity to develop such a model is to define all the connections and relationships among the operations and processes, and environmental conditions, and then use them to obtain an enterprise that is easily able to adapt to all circumstances. Thus, by the help of an enterprise model, enterprises can be managed based on a particular structure including analytical and conceptual components involving all relation points which provide rapid reaction to the change faced in the market whenever it is needed. Modern system thinking suggests that all enterprises should have a model on which their systems are integrated.

Enterprise models represent a functional working structure; functional components with input, output, and controlling relationships including behavioural aspects of enterprise activities; information structure, data flows, and information objects; physical resources and organizational aspects in an integrated framework. The detail level of the enterprise model depends on the purpose of modelling; the model components are similar and detail level only affects the number of schemes used in the model. Enterprise model creates an opportunity for the managers to look through the enterprise from the big picture providing ease of decide by considering all connection points and relationships. Besides managing the internal dynamics and environmental conditions, an enterprise model prepares all required infrastructure to establish an implementation of enterprise application systems, i.e., enterprise resource planning.

Building an enterprise model from scratch is a long lasting project that should be handled by a specialized team gathered from the process managers. The project manager of this team is a person who is well-trained about enterprise modelling, system analysis and design applications. This person is generally employed with the job title of *enterprise engineer*. Enterprise engineers coordinate all activities related with enterprise and system architectures.

An enterprise architecture is in parallel with a system architecture. The enterprise architecture takes some of its sources from system architecture concepts and specializes for enterprise modelling. The first applications in enterprise modelling did not consider pre-structured architecture during process modelling, but by the time being, enterprises have come to gather in a project to develop enterprise reference architectures. Thus, enterprise modelling not only provides integration within a single enterprise, but also a common language among the enterprises which are in touch with the same industry or same supply chain. After improvements and experiences, these architectures are standardized in international platform. Enterprises use the descriptions and definitions in these references to develop their own models. IDEF*, CIMOSA, GERAM, GRAI, ARIS are the suggested reference architectures in standards based on the characteristics of the enterprise. The details of these architectures are represented in the following chapters. In general, enterprise architecture modelling follows the similar phases to a software design process including requirements modelling, design, and implementation.

As in software design or in any design process from the broader perspective, any product, service, or process is designed with respect to the needs of target users or customers. Therefore, target users should be defined, observed and analyzed as the first stage of any design process. This phase is called *requirements modelling* and the tools and methodologies in this field are defined as *requirements engineering*. Potential target users in enterprises are employees, middle and senior managers, suppliers, customers, and all the other institutions which are in corporation with the enterprise. After the target users are selected in a requirements analysis process, they and their expectations are analyzed based on a user verbatim and observations to define their problems and needs underneath these problems. Finally, clarified needs are transferred to design characteristics. However, enterprises are different from the regular product/service designs in structure and scope. Enterprises are large and complicated systems performing operations in accordance with pre-defined goals. Enterprise goals are determined with respect to the needs in the enterprise and so enterprise goals should be reflected to the enterprise architecture, and finally to the enterprise model. Thus, requirements modelling is an important phase to be considered within the architecture.

Many tools and methodologies are defined in the scope of requirements engineering, e.g. use cases, unified modelling language (UML), Software-QFD, and text mining. Even if these tools and methodologies are first developed for software design processes, enterprise engineers frequently use some of them for enterprise modelling. Requirements analysis and modelling methodologies differ from each other in detail levels of analysis and modelling schemes. Some of them concentrate intensively on requirements modelling rather than analysis, e.g. UML, some of them are used only for analysis, e.g. text mining. Even if there exist hybrid applications of these methodologies and tools, modified models are also developed including detailed requirements analysis and modelling such as Software-QFD.

Modelling of user requirements is the first phase in enterprise modelling. Requirement engineering aims to discover user requirements that solve business problems. It also sets objectives as a measure of success in solving these problems, and determines constraints that limit the solution. Requirements are gathered from users verbatim and converted into models for validation. Requirement modelling with its analysis and representation phase is the process of delimiting the system and defining the functionality that the system should offer. A requirements model can behave as a contract between the developer and the user of the system, and thus forms the developer's view of what the customer wants. In this study, requirements analysis refers to the discovery of user needs and understanding their needs so that they are converted into design characteristics, while modelling refers to their representation to ensure their validity.

There exist many techniques for requirements modelling and analysis used in software development and process design. Yet, semiformal and formal techniques are not comprehensive enough for enterprise modelling requirements. Enterprises have their own stakeholders. They create expectations, constraints and limitations for the enterprise, and thus for the enterprise model. Therefore, modelling user requirements plays a key role, and is the first phase of the enterprise modelling process. Requirements modelling studies in the literature focus more on their representation than their discovery and analysis. However, the analysis phase is also important, and to be carried out as a process in which clear/hidden and structured/unstructured requirements are identified and analyzed.

Enterprises are integrated systems deploying their goals and strategies to perform their operations according to these goals. An enterprise structure emphasizes that the enterprise model should be well-matched with the enterprise goals and objectives. These goals and objectives cover not only the goals of the top management but also the goals of all entities interacting with the enterprise. A successful application and software integration in an enterprise needs a model-driven management, which is provided by the enterprise model. That is, this integration requires an enterprise modelling level that in turn needs a requirements analysis based on the enterprise goals and objectives.

The suggested frameworks in the literature can only be the starting point of requirements analysis, and a quantitative method should be developed for a successful conversion process. Long term and process goals should be not only classified, but also prioritized. Furthermore, prioritized goals should be analyzed with respect to the relationships with the short term process goals. After the final goal statements are obtained, these goals should be translated into the modelling constructs, i.e., functional, informational, and organizational characteristics. Therefore, there exists a gap between the structured goals of the enterprise and the general aspects of enterprise modelling because of the absence of any technique that translates the goals and characteristics of an enterprise into modelling aspects and constructs.

This thesis study aims at developing a novel approach for requirements analysis and modelling phase of enterprise modelling by considering the importance of enterprise modelling and integration, and the current status of literature. The study synthesizes enterprise modelling, requirements analysis and modelling, and QFD concepts and proposes an approach based on Modern QFD which is modified for enterprise modelling characteristics. QFD is a well-known approach frequently used in product/service design, and also modified for software design process. Since QFD has been employed in product design and software development successfully, it can also be extended to enterprise modelling to improve these approaches mentioned above. QFD involves not only requirements gathering (collecting, classifying, etc.), but also the design process from the inception to marketing through some structured forms with valid information. If the enterprise model itself is considered a product, QFD can be applied to its design and creation.

The closest QFD model for requirements analysis and modelling phase of enterprise modelling is Software-QFD. However, Software-QFD does not carry some of the characteristics related with enterprise architectures. Therefore, QFD can be modified for enterprise modelling to benefit from its characteristics by reflecting the architectural characteristic and modifying the content of its matrices. Finally, a novel approach has been developed based on Modern QFD by modifying tables and matrices with respect to the enterprise architectures, and this approach is defined as *Enterprise-QFD*.

Enterprise-QFD, developed based on Modern QFD by modifying existing parts and adding new matrices with respect to the enterprise modelling, totally differs from the QFD techniques that are developed for product/service or software design. Enterprise-QFD first collects the voice of users and analyzes and classifies them by converting them into clarified need statements, then finds out if the goals are related with these needs or not. This phase is called preparation phase before the quantitative analysis within Enterprise-QFD. The quantitative analysis starts with prioritization of long term goals of the enterprise. Enterprise-QFD employs Fuzzy-AHP (Analytical Hierarchy Process) technique for this purpose. Prioritized goals are handled to the first QFD matrix where the long term goals are converted into the process goals, thus the matrix chain is started. In the other phases, process goals are converted into processes, and then modelling constructs, i.e., functional, informational, resource, and organizational characteristics, respectively. Thus, desired modelling constructs are obtained through the integrated QFD matrices analyzing the relationships, benchmarking with the competitors and planning about the future, conflicts, technical challenges and advantages of requirements, as long as the adequate data are collected. Consequently, design targets of each requirement are determined according to the final importance values. The evaluation measures are calculated as ratio scales for each evaluation using Fuzzy-AHP, and importance values are obtained as linear distribution of the evaluation values on requirements. For each requirement, a measurement unit and target is determined at the end of each matrix. At the end of the Enterprise-QFD phases, requirement characteristics of each modelling construct is defined with the importance values.

With its integrated functions and stepwise application phases, Enterprise-QFD handles the requirements analysis phase as a whole project starting with the enterprise goals and ending when all inputs are ready to convert them to formal reference architectures of enterprise modelling. Thus, Enterprise-QFD fulfils the gaps between the requirements model and the corresponding enterprise architecture.

Enterprise-QFD, as a requirement analysis approach proposed especially for enterprise engineers, can manage all requirement analysis and definition phase of enterprise modelling. Enterprise-QFD can support all enterprise reference architectures and modelling components during the requirement definition phase of the modelling by presenting an analytical approach for requirement analysis and definition even for the most complex references.

In the scope of the study, Enterprise-QFD is applied to a small business company processing steel products with real evaluations and the findings to show the usability of the method. The results show that Enterprise-QFD generates the infrastructure for further modelling of enterprise architectures concerning both functional characteristics of the enterprise and needs of stakeholders. A toolbox is presented that provides a user friendly application platform for Enterprise-QFD design utilizing MS-Excel VBA tools. After the requirements are analyzed and modelled by Enterprise-QFD, the findings are transferred to the requirements model of CIMOSA.

The arrangement and subject flow of this thesis has been determined in accordance with the components and basis of Enterprise-QFD. After this introduction, firstly, enterprise integration, enterprise modelling and enterprise engineering concepts are defined in chapter 2. After the basic concepts are defined, chapter 3 introduces the most commonly used enterprise reference architectures and their application areas in the literature. Requirements analysis and modelling described as the first phase of enterprise modelling are explained with concepts, tools, methodologies, and current studies in chapter 4. Chapter 5 is written on QFD as a well-known requirement analysis methodology where it is examined and explained according to its historical development, application areas and current modifications. AHP and Fuzzy-AHP which are applied for the evaluations of QFD are also explained in this chapter. Chapter 6 is a special part written on a novel framework where concepts in chapter 2 through 5 are synthesized on a novel analytical approach, proposed as Enterprise-QFD, to manage the requirements analysis and modelling phase of enterprise modelling. Chapter 6 also presents the user friendly toolbox design for Enterprise-QFD and application of this approach in a small sized company with CIMOSA representations. Chapter 7 concludes this thesis study by summarizing the theoretical framework and proposed model, the application findings, and benefits. The future work is then discussed in the conclusion part.

CHAPTER TWO
ENTERPRISE MODELLING AND INTEGRATION

2.1 Enterprise Integration

Enterprise integration and modelling is the re-engineering of business processes and information systems to improve teamwork and coordination across organizational boundaries, thereby increasing the effectiveness of the enterprise as a whole. Although there are companies reporting dramatic improvements in cost, quality, and schedules, there is also disappointment reported in many corporations due to unmet expectations (Nagarajan, Whitman, & Cheraghi, 1999)

Global competition demands shorter lifecycles and customer values. The concept of achieving integration among functional requirements, resources, organization and information is still upheld as a critical element for the success of enterprise. Integration is never-ending process. Both internal and external environment can change overtime. The enterprise should react to these changes. Providing the right information at the right time requires explicit knowledge of both the information needed and created by the different activities in the enterprise operation (Ortiz, Lario, & Ros, 1999).

The enterprise model will allow more consistent modularization so that enterprises can interchange pieces. The models will ameliorate the need to develop the entire system at one time. Simulation will be possible allowing evaluation of inter-operation with inter-enterprise entities and evaluation of systems with differing granularity. Enterprises will be able to plan migration paths more effectively. Because information will be a separate asset, changing applications will be possible without re-entering information about the products and processes unnecessarily. Enterprises can define paths to make the product and process information tie logically into enterprise goals, strategies, capabilities, and business rules. The models should be scalable so that a high-level model is essentially the same as a lower-level model. That is, use the same modelling constructs for all levels. Enterprises need to manage all systems and tools as an integrated management system and enterprise modelling provides the infrastructure for such an integrated platform. The current need for enterprise-wide integration of business organization can be explained by several reasons. Some of the most relevant ones are:

- The need to keep business operations aligned with strategy.
- The need to share enterprise information, (data, used for decision making).
- The need to interoperate, i.e., the need for the different systems that exist in the enterprise to be able to work with each other, even across organization boundaries (extended and virtual enterprises)
- The need to generate models and tools which let the users estimate the impact of the decisions taken in view of the globalisation of markets and the need for fast and effective response of enterprises.

Enterprise Integration (EI) consists in facilitating the material, information, decision and control flows throughout the organization, linking functions with information, resources, applications and people, with the aim of improving communication, cooperation and coordination in the enterprise, in order to manage the enterprise to behave as a whole and operate according to the strategy of the enterprise (Ortiz, Lario, & Ros, 1999)

To reach these objectives, all levels of enterprises must be considered, from the most strategic to the most operative ones. They must evolve in a coherent framework, which enables the actions and decisions to be made at each level of the enterprise. Figure 2.1 represents these levels:

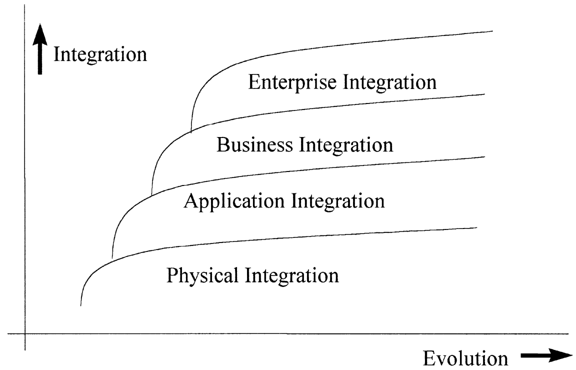


Figure 2.1 Enterprise integration evolutions.
Reference: (Ortiz, Lario, & Ros, 1999)

It is necessary to consider all aspects relating to enterprise strategy and business processes, as well as to the modelling, construction and execution of these processes to progress towards enterprise integration. Additionally, the consequences of the enterprise modelling program on the human resources and the impact of the human resources on the success possibilities of the program must not be ignored. To cover these aspects and make a step forward on the path towards EI in a coherent and effective way, it is necessary to provide the enterprises with three necessary elements: a methodology, architecture and tools. Therefore, these three elements are the ones which make up the framework in Figure 2.2.

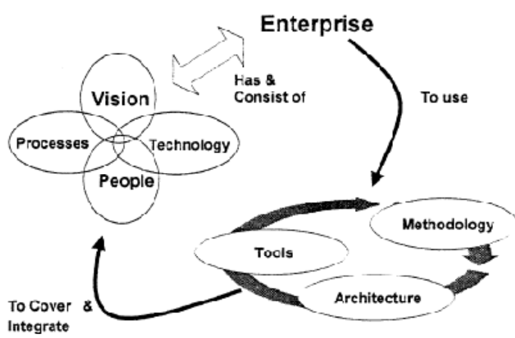


Figure 2.2 Framework for enterprise integration.
Reference: (Ortiz, Lario, & Ros, 1999)

The approach presented is based on the necessity to cover the whole life cycle of a business entity, from its identification to its disposal:

- Taking into account the strategy of the enterprise (vision),

- Applying it to the business processes, as they are the ones which provide the higher congruency and integration between the activities developed in the enterprise (Hammer & Champy, 1993); (Davenport, 1993).
- Using structured techniques,
- Developing enterprise applications, and
- Keeping in mind the role played by humans and enterprise technologies.

One possibility to develop this vision was to create these elements from scratch. But this would have been a major mistake as it would have discarded all the existing work and know-how developed by numerous R&D projects.
Chapter 6 represents a framework including similar components with Figure 2.2 but proposes methodologies for the components.

2.2 Conceptual Framework of Enterprise Modelling

Today new business forces are demanding of business enterprises to adopt more formal knowledge management. Rapid organizational changes, knowledge-intensity of goods and services, the growth in organizational scope, and information technology have intensified organizational needs for knowledge. In addition virtual organizations that are made up of complementary allied entities place greater demands on knowledge sharing (Ruggles, 1995).

Unstructured business knowledge is important for a company's performance, but cannot be systematically used and is not an asset a company can own. Clearly there is a need for support in terms of conceptual frameworks for structuring and managing enterprise knowledge so that it is clearly defined, controlled, and provided in a way that makes sure that it is available and used when needed. To this end, the role of conceptual modelling is critical. Loucopoulos & Kavakli(1999) shows how conceptual modelling fits in the wider spectrum of enterprise knowledge management by defining the requisite methodological framework. Allied to enterprise knowledge modelling is the larger issue of *enterprise change management* itself. Enterprise change management needs for enterprise model provides the required analysis input for knowledge management. Thus, for the construction and sustainability of relevant knowledge management, enterprises should analyze and model its needs, and then figure out its modelling architecture on which the knowledge model would integrate.

2.2.1 Definitions

Enterprise modelling is defined by many authors from the different perspectives. Followings are the most common definitions which are used in the literature.

"An enterprise model is a computational representation of the structure, activities, processes, information, resources, people, behaviour, goals, and constraints of a business, government, or other enterprise. The role of the enterprise model is to achieve model driven enterprise design, analysis and operation. It can be both descriptive and definitional spanning what is and what should be within the enterprise. From a designer's perspective, an enterprise model should provide the language used to define the enterprise different from the others; from the operations perspective, the enterprise model must be able to represent what is planned, what it might happen and what has happened (Fox & Gruninger, 1998)." This definition is the basic definition to determine the general boundaries of the enterprise model, and detailed terminology can be added to see all the aspects and promises of enterprise modelling.

"An enterprise model is a symbolic representation of the enterprise and the things that it deals with. It contains representation of facts, objects and relationship that occur within the enterprise. Enterprise assists the enterprise engineering by helping to represent and analyze the structure of activities and their interactions" (Liles & Presley, 1996).

"Enterprise model contains both static and dynamic views of the enterprise" (Pardasani & Chan, 1992), and all other aspects of enterprise models can be categorized under these titles.

“An enterprise model is a model of what an enterprise intends to accomplish and how it operates. It identifies the basic elements and their decomposition to any necessary degree. It also specifies the information requirements of these elements. It provides the information needed to define the requirements for integrated information systems. This feature is used to improve the effectiveness and efficiency of the enterprise” (ANSI/MEA, 1994). This definition only comprises the information aspect rather than all other basic aspects such as functional, organizational, and resource views.

“An enterprise model is a concise description of what an enterprise does to operate. In this context, an enterprise model usually means either a series of graphical representations or highly structured textual description. An enterprise model is a representation of the enterprise itself and how it works. A well-designed enterprise model provides both a broad view of the enterprise, and a means to isolate and review specific portions of interest. It may not be as formal as mathematical model. It may take the forms of a series of diagrams, a collection of tables or matrices, and a sequence of statements in a structured or stylized language, or some combination of these and other descriptive forms. It may fall into the category of either descriptive models, which describes what the operation of the enterprise is like or prescriptive models, which describes not the way the things are, but the way the management would like it to be. It may also contain many other sub models, such as entity or process model” (Eiric, 1992). This detailed definition provides how the enterprise models should be. It emphasizes the fact that enterprise models need not always represent the entire enterprise; it may also focus on specific areas.

“An enterprise model is a structural description of an organization in terms of variables and essential relationships. It reveals the basic structure of an organization, explains how it functions, and predicts its future behaviour. Typical applications include diagnosing the performance of an organization, predicting the behaviour of an organization over time, testing the implications of theories about organizations and supporting strategic business decisions” (Ba, Hinkkanen, & Whinston, 1994).

Different from the other definitions, the application perspective is added in this definition.

"An enterprise model may be anything from a factory blueprint, to a model built form a static meta-model template, or a dynamic model driven by a collaborative, distributed modelling tool. The modelling should be preferably be supported by visual libraries of model templates and hierarchies of meta-data, and a guiding methodology model. The four main dimensions of any enterprise are: product and services, organization and people, processes and work items, and system and tools" (Lillehagen & Karlsen, 1996). This definition again focuses on information systems with related to the other aspects of the enterprise.

"Enterprise model shows the basic, fundamental functions, processes or activities of an enterprise or an organization, often reduced to just one, two or three key activities on top, and then decomposed to sub-activities to the desired level of detail" (FAA, 1995). The definition is similar to (ANSI/MEA, 1994) describing the representation of the enterprise model and its decomposition except focusing on critical activities.

"Enterprise model is one or more models that is used to document the process and data for an organization, business or enterprise and serves as the point of planning and integration for all information systems management. The enterprise process model represents the major processes of an organization. With the exception of the level of detail, the techniques used in building enterprise models are the same as those used to construct application data and process models" (Smith, 1996). The definition depicts an enterprise model as one or more models for documenting enterprise-related processes and data, which are used in the information systems management.

"An enterprise model is a graphical or computational representation of enterprises; aiming to promote communication and understanding of business processes while at the same time provide a framework for assessing changes and making forecasts" (Berio & Vernadat, 1999). This definition also brings the notion of framework into picture; it also identifies using models for assessing changes and forecasts.

An enterprise model is a consistent set of special purpose and complimentary models describing the various facets of an enterprise to satisfy some purpose of business users. An enterprise model usually consists of, but is not limited to, product models, resource models, activity models, information models, organizational models, economic models and decision making models (Vernadat, 1996, p.23). This definition is written based on all type of models from the different detail levels and in contradiction with the definitions that consider the enterprise model as a macro approach and meta-model. However in the details of enterprise modelling, there is a concentration on the macro-model before the detailed levels of modelling. Vernadat (1996) also implies that an enterprise model is one representation of a perception of an enterprise. It can be compared of several sub-models including process models, data models, resource models and organization models. The content of an enterprise model is whatever an enterprise considers important for its operation.

"An enterprise model is an abstraction that represents the basic elements of an enterprise and their decomposition to any necessary degree. It also specifies the informational requirements of these elements, and provides the information needed to define the requirements for integrated information systems" (ISO, 1998a). This definition is similar to (ANSI/MEA, 1994) but does not consider the other basic elements within an enterprise.

2.3 The Purpose of Enterprise Modelling

Enterprise modelling plays a central role in enterprise engineering by mediating between multidisciplinary viewpoints of system designers and system users. It should be managed as any project handled by all the process owners within the enterprise by bringing the voice and experience of end-users at all stages of the system design. Different enterprises and/or different end-users may concern different utilities which can be obtained from enterprise modelling. Thus, there is not a unique list for purpose of modelling (Vernadat, 1996, p.69). However, some common declarations can be observed as follows:

- to better represent and understand how the enterprise (or some part of it) works,
- to capitalize on acquired knowledge and know-how for later reuse,
- to rationalize and secure information flows,
- to design (or redesign) and specify a part of the enterprise (functional, behavioural, informational, organizational, or structural aspects),
- to analyze some aspects of the enterprise (economical, organization, qualitative, quantitative, facility analysis),
- to simulate the behaviour of some parts of the enterprise,
- to make better decisions about enterprise operations and organization,
- to control, coordinate, or monitor some parts of the enterprise, i.e., processes.

Enterprises are complex systems in terms of number of entities involved, things to do, decision variables to be considered, and processes to be controlled. The complexity comes from the potential number of interactions among the processes or objects, and occurrences of unexpected events (internal/external) that affect system operations. Models and performance indicators used by top management must be based on the aggregation of low-level information. Models used by middle management are more detailed but have a narrower focus. This levelling goes on along the hierarchical structure of the organization down to the operational level where the model reaches to its full complexity. The important issue in managing manufacturing enterprise complexity is to find out a rational way of managing the hundreds of daily business processes involving thousands of operations, accessing and processing of huge data, papers, and resource usage. However, the condition in the market where the competency is very hard forces the enterprises for rapid adaptation and change. One way to reduce the complexity and contradiction is to follow hierarchical problem solving approach which is considered in enterprise modelling as system decomposition. Things are better controlled if they are better understood, and if an easily interpretable representation is available. This is why enterprise management needs a model of an enterprise, performance indicators, and decision rules. These necessities can individually be a component of an enterprise model. The most common components are explained in section 2.4.

2.4 The Components of Enterprise Modelling

Efficient daily enterprise management and operations require at least good knowledge of current situation and the target objectives; timely process coordination; reliable information system structure and management; robust resource management policies; and an adequate organizational structure. According to the control theory, any time a system needs to be controlled or analyzed, a model is required. Models are also required for decision making activities. This is especially true for integrated enterprises for which model integration is a central issue (Vernadat, 1996, p.80).

The common specification of the definitions given in section 2.1 is that the modelling of the enterprise means to represent which activities are handled and managed within the enterprise concerning its behavioural characteristics. Table 2.1 outlines the activities in an enterprise by classifying them as ‘what’, ‘how’, and ‘do’ activities.

Table 2.1 Major enterprise activities

Example What, How, and Do enterprise activities (ISO 14258)			
	What activities	How activities	Do activities
Plan and build phase (e.g. before sell/buy title transfer) Use and operate phase (e.g. after sell/buy title transfer) Dispose and recycle phase (after product is no longer useful	Develop goals Define strategy Define product needs Define support needs Define use Define recycle/dispose needs	Develop requirements Define concept Design product Plan to produce product Plan to support product Define use /support requirements Define dispose requirements	Procure parts Produce product Test product Ship product Use the product Support product Recycle product Dispose product

Reference: (Weston, 1999)

The majority of enterprise modelling techniques provides concise descriptions of what an enterprise “does” in order to operate. To this end, they usually involve two kinds of sub-models. An entity (or data, or information) model and a process (or functional) model (ICEIMT, 1992). For example IDEF0 diagrams (IDEF0, 1993);

DFDs (DeMarco, 1978) or workflows (Swenson & Irwin, 1995) are widely used to describe enterprise processes, while entity relationship based diagrams (Chen, 1976) are in common use for enterprise data modelling. However, these enterprise models ignore important topics like: what is the social and organizational structure of the enterprise; what are the roles of enterprise agents; what are the reasons, objectives, motivations that define the enterprise structure and processes. In recent research studies, enterprises handled as a whole including not only the functional characteristics but also the organizational, informational, and resource characteristics with their behavioural aspects.

To model the functionality and behaviour, one needs to model resources and temporal events, and then when processes and information flows are modelled, these flows should be allocated to some organization units which have control on them (AMICE, 1993). Thus, an enterprise model usually consists of (not limited to) following components (Vernadat, 1996, p.72):

- **product models**, which are used to represent geometric and non-geometric features as well as design details of products and their parts made in the enterprise throughout the product life cycle;
- **resource models**, which describe characteristics, layout, management policies, and possible actions of pieces of equipment as well as their configuration to perform enterprise activities;
- **activity models**, which indicate the set of operations (or actions) to be performed to execute enterprise activities and do the work;
- **information models**, which describe the structure and the relationships of data and information elements of the enterprise information system;
- **organizational models**, which document the organizational structure of the enterprise in terms of plants, departments, cells, stations, and work centers as well as authorities and responsibilities assigned to each decision level;
- **economic models,** which provide a cost-oriented analytical view of the enterprise used to evaluate the cost-effectiveness of the various parts of the enterprise;

- **optimization and decision-making models,** issued from operations research and control theory and used by decision support systems.

Most of these models can themselves be broken down into more detailed sub models. Enterprise Modelling integrates the components mentioned in this section in a pre-defined scope and boundaries within the enterprise. Section 2.5 explains how the scope and the limits of the enterprise model can be described.

2.5 Scope of Enterprise Modelling

Basically, enterprise modelling is concerned with modelling the **what, how, when,** and **who** aspects of an enterprise. What essentially refers to operations performed and objects processed in the enterprise. The how defines the enterprise behaviour, i.e., the way things are done. The when enforces the notion of time as being an essential component of the model. It can be associated to events representing a change in the state of the enterprise at a certain time. The who concerns the resources or agents; the enterprise performing operations of the business processes? Of course, the **how much** (economic aspects) and where (logistics aspects) are also important aspects of an enterprise to be considered.

Based on this assumption, four basic aspects to be modelled in an enterprise are defined in the survey paper on process modelling (Curtis, Kellner, & Over, 1992):

- functional aspects describing what has to be done;
- behavioural aspects defining how and when something has to be done;
- informational aspects defining what data are used or produced and their relationships;
- organizational aspects indicating who has to do something and where.

An enterprise is by nature a complex dynamic system. From the point of view of integration, various essential aspects of an enterprise need to be modelled, either to analyze or to control the system. These include but are not limited to (Vernadat & Zelm, 1993):

- enterprise functionality and behaviour in terms of processes, activities, basic functional operations, and triggering events;
- decision-making processes, decision flows, and decision centers;
- products, their logistics, and their life cycle;
- physical components or resources, e.g. machines, tools, storage devices, or transportation means, their logistics, capabilities, capacities, and layout;
- applications (i.e. software packages) in terms of their basic functional capabilities;
- business data and information and their flows in the form of orders, documents, data items, data files, or complex databases;
- enterprise knowledge and know-how,e.g. domain-specific knowledge, rules of thumb, specific decision-making rules, internal management policies, international regulations, etc.;
- human individuals, especially their qualification, skills, roles, and availability;
- organizational structure, i.e., organization units, decision levels, decision centers, and their relationships;
- responsibility and authority distribution over each of the previous elements;
- exceptional events and reaction policies to these; and
- time, because an enterprise is a dynamic system.

Because the description of all these enterprise elements cannot be fully represented in just one model, it usually results in different, more or less interconnected, overlapping models. We have previously mentioned the product models, process models, functional models, information models and their databases, knowledge bases, resource models, configuration models, organization models, decision models, or economic models, etc.

Enterprise models are constructed using the components in section 2.4 in a pre-defined scope defined in this section. During practical implementations of enterprise models, some principles have been raised to manage the further modelling processes. Section 2.6 explains these standardized principles.

2.6 Principles of Enterprise Modelling

In addition to general modelling concepts including definition of purpose, boundary, aspects, and the level of the model, enterprise models are constructed concerning the following principles (Vernadat, 1996, p.81):

1. Principle of separation of concerns: Due to its inherent complexity, it would be unrealistic to consider an enterprise as a whole. It must, therefore, be analyzed piece by piece, each one corresponding to *n* existing separate functional area or domain (such as a product design process, master production planning, or a manufacturing plant). This is a way of breaking down the complexity of enterprise models.

2. Principle of functional decomposition: Enterprises are complex dynamic systems mostly defined by their functionality. Major functions are structured into sub-functions, sub-functions into sub-sub-functions, and so on, according to the breakdown of business objectives into sub-objectives, and then sub-sub-objectives, etc. All enterprise modelling methods provide such a stepwise-refinement approach as originally systematized in SADT (Ross, 1977).

3. Principle of modularity: To facilitate management of change, models must be modular, i.e., made of an assembly of compatible building blocks so that the model can be built on a 'plug and play' basis. This is a second way of dealing with enterprise model complexity and it makes model maintenance much easier.

4. Principle of model genericity: Many activities or components of an enterprise exhibit identical or similar properties although enterprises are generally different. It is therefore important to define standard building blocks as generic classes to factor common descriptive attributes and behaviours. These classes can then be adapted or specialized in the modelling of peculiar components or applications. Key concepts of objects, object classes, and inheritance as proposed by object-oriented approaches

provide the necessary underlying principles and guidance in this respect. This is another way of handling enterprise modelling complexity.

5. Principle of reusability: To reduce modelling efforts and increase model modularity, predefined building blocks or partial models must be reused and customized to specific needs as much as possible when modelling new parts of the system. This refers to customization. This is another way of breaking down enterprise model complexity and of reducing model development cycle times.

6. Principle of separation of behaviour and functionality: Enterprise behaviour should not be confused with enterprise functionality if organizational flexibility has to be enforced. Enterprise functionality concerns the 'things to be done' by functional entities, while enterprise behaviour defines 'how things are done' (AMICE, 1993). A clear distinction between the two in the model and its implementation will allow modification of one without impacting the other, and vice versa.

7. Principle of process and resource decoupling: Similarly, it is important to separately consider the things being done (i.e. processes) and the agents performing them (i.e. resources) to preserve operational flexibility. The mapping between the two is a scheduling problem particularly critical in manufacturing systems and project management. This mapping can be done ahead in time (traditional planning and scheduling problems), or on-the-fly at run time.

8. Principle of conformity: This principle is the most difficult one to address. It deals with syntax and semantics of the model and concerns the ability of the model to really and accurately represent what it is supposed to model. Modelling constructs of the modelling language must therefore be provided with a clear syntax and semantics which must be minimal for the application domain covered. In other words, the modelling language must be consistent and non-redundant.

Strict adherence to the principles of model genericity, modularity, and reusability makes it possible to build CIM systems tailored to user needs from standardized predefined building blocks and software modules stored in libraries or available in the marketplace (Naeger & Rembold, 1994).

In addition to these, additional principles are considered as follows (Ward & Mellor, 1985):

9. Principle of model visualization: To easily communicate models, the modelling approach should be supported by a non ambiguous and simple graphical formalism.

10. Principle of simplicity versus adequacy: A prime characteristic of any modelling language is to be rich enough to express what needs to be expressed. However, on the one hand a language with a few words cannot correctly model complex subjects, and on the other hand a complex language may require too much effort first to be learnt and then to be correctly mastered and used.

11. Principle of management of complexity: Any system modelling language must permit the representation of systems of arbitrary-great complexity.

12. Principle of rigor of representation: The model must neither be ambiguous nor redundant nor serve as a basis for verifying properties, analyzing behaviour, or simulating the system modelled.

13. Principle of separation of data and control: A modelling language that is to be adequate for real-time systems must be capable of separating the data needed by a process from the control that actually makes the process operate. The process will not simply be triggered by data availability but by some events. Thus, the control must be modelled as well as data.

Very few modelling techniques and methods for enterprise modelling correctly address all these principles as itself. Nowadays interoperability of the models is discussed so that their capabilities can be used together and the modelling principles are covered (Vernadat, 1996, p.82). The following section describes how an enterprise modelling process can be handled.

2.7 Enterprise Modelling Process

Enterprise modelling consists of understanding the essential features of a system and recording them systematically. It can be seen as the process of building models of whole or part of the enterprise from knowledge about enterprise, previous models, and/or reference models. A modelling process is a set of activities to be followed for creating one or more models of the defined universe of discourse and given purpose.

"Enterprise modelling starts with the capture of user requirements in the form of business descriptions and business issues (e.g. explanations from user interviews, sketches of processes, examples of data screen, samples of data and documents, etc.). The process with a formalized description of enterprise operations defines what has to be done in the enterprise, how it will be done, and by whom in specific contexts such as specific conditions and situations" (Vernadat, 1996, p.85).

In order to model an enterprise, other models, i.e., partial or reference models stored in libraries, can be used as well as domain ontology. The process transforms an enterprise into a set of models representing different aspects of the enterprise and a new set of ontology for the domain. This process is managed by the use of a methodology and needs criteria to stop the process as well as the metrics to qualify the models (Petrie, 1992). Figure 2.3 represents the overview of the enterprise modelling process.

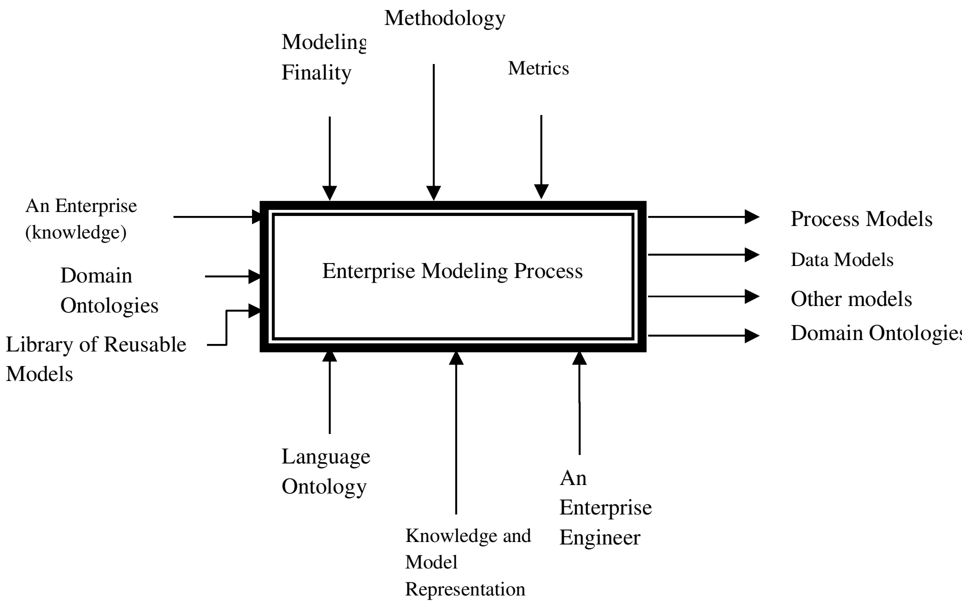


Figure 2.3 Overview of enterprise modelling process.
Reference: (Petrie, 1992) in (Vernadat, 1996, p.85)

In accordance with the overview given in Figure 2.3, the basic steps of enterprise modelling is represented in Figure 2.4.

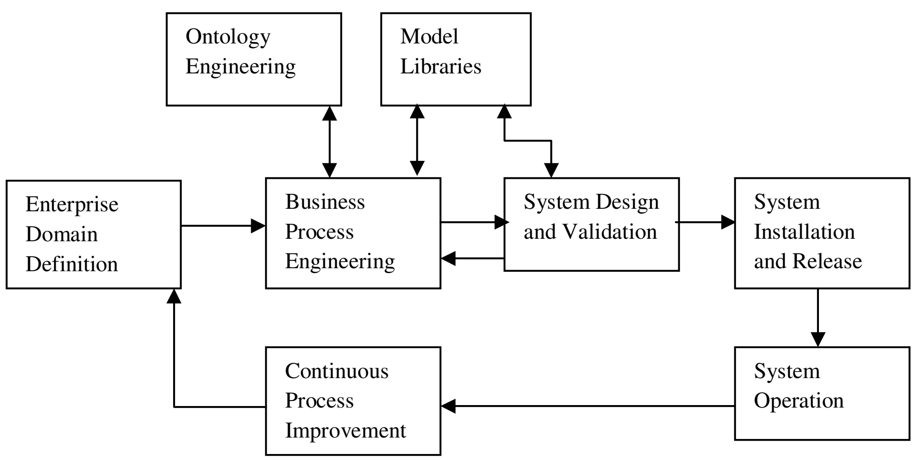


Figure 2.4 Basic steps of enterprise modelling process.
Reference: (Vernadat, 1996, p.86)

Figure 2.4 provides a more detailed view of the modelling process, insisting on the feedback loop to business process engineering via continuous process improvement, and suggesting that the enterprise modelling process is a never ending process. Indeed, the enterprise model must be kept in line with the evolution of the enterprise, and can even be used to anticipate the enterprise changes.

Information collection by group meetings consists of forming groups of people in the enterprise that must be trained to the modelling technique used. They discuss until they reach a consensus view, which is the basis for the model development. Usually, this approach takes time and is very costly because of the number of people involved (who cannot do their regular work during this time) and time spent (several meetings required).

Information collection by interviews consists of sending experience: analysts in the enterprise who collect user descriptions as well as samples of documents or data used directly from users. Compared to the previous approach, the latter takes less time and is less costly but provides less exposure of the users to the model.

Collection of data and/or data itself as the input of enterprise modelling is generally handled by the methods explained in the previously. But they have some disadvantages about generating the reliable data and the time to get them. Thus, a systematic methodology including reliable metric can be used for gathering and analyzing the voice of user so that the engineers obtain the reliable data in a short time. In the scope of this thesis, such a methodology is proposed based on Quality Function Deployment.

A model is useful if the users consider it as an adequate model. This is the fundamental point that a model is useful only if it is used. It will be used only if it is practical and if it makes sense to end-users. Similar issues apply to modelling techniques. They will be accepted by users as a tool if they are simple to understand, easy to use, computer-supported, and if they provide a realistic image of the reality. This explains the failure of many approaches proposed in the past, and the difficulty of some of the current techniques. The difficulty for tool builders is to develop sophisticated modelling and analysis environments which hide this complexity and have a friendly user interface, good graphical representation, and the language of the user while at the same time offering powerful analysis and simulation capabilities. Menu-driven window systems, with the use of a mouse implemented on top of object-oriented programming environments certainly form the basic platform configuration necessary to reach this goal (Vernadat, 1996).

The use of enterprise modelling concept and the way and the architecture used for implementation is examined through a survey study. According to this study, enterprise models are used to answer a wide variety of questions in a wide variety of enterprises. The primary research question of this survey was the use of enterprise models with a particular focus on the three dimensions of living models. It was not expected that half of the respondents would claim their enterprise models encompassed their entire division, multiple divisions, and even multiple enterprises. It is encouraging to see that enterprise models are being used on such a wide scope. The pervasiveness of enterprise models was not as large as was expected. Of the respondents, 75% claimed that their models did not receive information from the enterprise more frequently than quarterly. The same is true for how often the models provided information to the enterprise. How often the models are updated also posed some concern, as 75% do not update their models more than five times (although, 32% update the model three to five times). It was difficult to get a firm grasp as to how many models are used, as most respondents did not know the use of models beyond their own experience (Withman & Huff, 2001).

Enterprise modelling frameworks and approaches differ, but what they are intended for the possibility to understand the application enterprise appropriately (Greenbaum & Kyng, 1991). The basic keyword here is “to understand” which should consist of elements of the enterprise and its relations with different aspects, its role and behaviours to any change in environment. In this regard, models should have an explanatory capability beside the representation. The explanatory capability has the major importance from the information system development. In the context of information systems development, the following three systems can be reflected by the enterprise model (Kirikova, 2000):

- the application enterprise;
- the system of the requirements;
- the information system

All three systems are mutually related and add onto each other with respect to explanatory dimension. Each of them (if present in the modelling framework) can be represented by one or more sub-models of the enterprise model in use. Table 2.2 is one of the first perspectives of the explanatory principles in the literature (Dahlbom & Mandahl, 1994).

Table 2.2 Explanatory capability of enterprise modelling

Explanatory principle	Interpretation of the principle by Dahlbom and Mandahl	Enterprise modelling methodology				
		Why?	How?	Who?	What?	Using what?
		Objectives	Activities and usage	Actors	Concepts	Inf. syst. requirem.
Material	Capital		Material	Non-human resource		
	Technology		Process			
	Personnel			Individual Organisat. unit Role		
	Basic education and skills of personnel			Individual		
	(Indirectly) systems of finance, laws, market, etc., in society at large	Rule	External process			
Formal	The way the business is organised formally		Process Material flow Information flow	Organisat. unit Role		
	The way the business is organised informally					
Efficient	Actual daily activities performed by members of organisation					
	[Planned daily activities]		Process	Individual		
Final	The ultimate goal of the organisation (mission)	Goal				

Reference: (Dahlbom & Mandahl, 1994)

As seen in Table 2.2, the capabilities in the principles are closely related with the hierarchical decision and management levels within an enterprise starting from the vision, mission, goals and ends in configurations including major functions and processes (Kirikova, 2000).

2.8 Enterprise Engineering

Needs and requirements for these technologies develop, hopefully, using activity, information, and dynamic behaviour modelling. The process of analysing the enterprise is perhaps the most important engineering activity because it enables a relatively deep understanding about what is really happening in the enterprise processes. From this understanding, together with the enterprise goals and strategies, comes justification for improvements to integration.

Managing the enterprise architecture and finally the modelling is handled by enterprise engineers.

"Enterprise engineering can be defined as the art of understanding, defining, specifying, analyzing, and implementing business processes for the entire enterprise life cycle, so that the enterprise can achieve its objectives, be cost-effective, and be more competitive in its market environment (Vernadat, 1996, p.30)."

The idea behind enterprise engineering is that enterprise systems can be engineered in a systematic way like any other complex systems. It includes industrial engineering approaches such as methods for business process definition, cost-based analysis, logistics, process design, resource selection, or manufacturing layout design, quality standards but adds techniques for workflow management, information system design and analysis, dynamic resource allocation and management, or design of organizational structures, etc. It is an interdisciplinary, large-scale effort carried out by cooperating teams of users, designers, analysts, and managers. Enterprise engineering is therefore at the crossroads of many disciplines concerned with the design, re-engineering, and continuous improvement of business processes (i.e. BPR and CPI) of manufacturing enterprises (Vernadat, 1996, p.32).

Enterprise engineering must rely on structured approaches (for which sound methodologies are still to be defined), and be supported by powerful computer-aided enterprise engineering (CAEE) tools (currently under development) to cover the whole system life cycle (Ladet & Vernadat, 1995). These tools will have a lot of functionalities and capabilities already offered by computer-aided software engineering (CASE) tools, but need to be significantly expanded in a number of ways for specific aspects of enterprise engineering such as resource management, organizational structure definition, analysis of concurrent processes, or event-driven model enactment in highly distributed environments (Aguar, Coutts, & Weston, 1995, p.62-83).

2.9 The Role of Standards in Enterprise Engineering and Integration

The characteristics of effective enterprise models probably are quite specific and fairly straightforward. It seems that the structuring and concepts used in enterprise models is a good area to constrain the enterprise representation. If we assume that the enterprise is model driven, it seems logical that standards constrain the end products of the representation of components in an enterprise. Innovators will continue to design enterprises by seeking optimum solutions. They will continually update and reorganize processes and the infrastructure. However, each process or component of the enterprise including technology and infrastructure technology will need the same things to inter-operate. This means that when modelling these components, if the information presented in the model views is consistently there; say, required by a standard, designers could connect enterprises or pieces of enterprises to other enterprises and operate effectively. Therefore, the structuring and concepts used in enterprise models appear to be a candidate for standardisation. To be able to link models from different sources, those models have to behave as a common model. This requires a meta-level semantic unification, which provides a common modelling language base and allows standard interfaces between different representation dialects used in the models to be linked (Kosanke & Nell, 1999).

Enterprise models provide a data-driven and model-driven enterprise with several capabilities. Whether or not the integrated enterprise operates in a hierarchical, deterministic mode or in a distributed, chaotic mode, the enterprise model will provide the operator or executive, human or machine, with a map of the enterprise and some knowledge of what functions the enterprise comprises, in what state they are, and what capabilities exist at any moment to accomplish an output. If the models conform to some established framework, enterprises can seek, evaluate, set up, and go more easily toward inter-enterprise as well as intra-enterprise commerce. With well-designed standards about enterprise representation models in place to provide a known environment to the developer, the risks of investing in an island of integration will be significantly reduced. If confronted with one of those islands, the technology required to interact with a standard environment will be a known quantity. A good standard will guide and constrain existing and emerging enterprise models so that resulting pieces of enterprises will inter-operate with each other and formulate migration strategies with confidence. The resultant environment will create a more confident investment climate for integration-technology related human and technical resources (Kosanke & Nell, 1999).

The domain of enterprise engineering and integration consists of hardware, software, communication protocols, information, frameworks, and architectures. There are things, the connections between the things, the information, and the information formats. With respect to enterprise representation, what level of concept should be standardised, from entire standard enterprises to standard names of things? Of what value are standards covering enterprise models, enterprise modelling, enterprise-reference architectures, or frameworks? Assuming that standardized enterprises and processes are not feasible, then at what level is a standard appropriate? Standardising the enterprises, parts of the enterprise, the products, the information transferred, and the processes, is probably not going to be a productive use of standard-making resources. What seems more usable is to standardise the interfaces between components and the formats and allow the tool builders to use these standards to design software and process within in a virtual enterprise. (Kosanke & Nell, 1999).

Several approaches oriented towards the improvement of the enterprise’s competitiveness are appeared, like total quality management, process reengineering, collaborations between enterprises, virtual enterprise, improvement of the availability of information, flexibility, and integration of customers and suppliers. These new tendencies and innovations in the fields of management and technology are always been handled in enterprises in an isolated and uncoordinated way. Thus the large promised "improvement expectations" are not accomplished (Chalmeta, Campos, & Grangel, 2001). Therefore, in order to achieve all the possibilities that these new and better methods and tools offer, an enterprise must "efficiently manage" all its elements, aligning and integrating them in order to improve the ability to work together in a "continuous improvement process" toward the accomplishment of the objectives and the strategy of the enterprise.

One area where standards are important to help with the enterprise engineering and integration work is in enterprise-process representation. The ISO standards group in this domain is TC184 SC5 WG1, *Industrial-automation systems and integration*, *Architecture*, *communications*, *and integration frameworks*, *Modelling and architecture*. WG1 is planning a family of standards that will help manufacturers, implementers, software developers, and other standard makers to create consistent environments in which the integration process can progress. To engineer and improve the integration level of an enterprise, WG1 can envision standards in four key areas: process representation, integrating infrastructure, a semantics-resolving utility and representation of human involvement. These are in addition to the basic standards required to assure compatibility among interacting hardware, software, communication protocols, and information format. The key areas require varying degrees of research and development to precede the standards work, and projects are being organised in some areas. WG1 is creating a road map of the enterprise-representation domain to help to plan and prioritise its work (Kosanke & Nell, 1997). Figure 2.5 identifies the available standards, relevant state of the art in standardisation, future work items, and related standards. The state of the art includes work done by the European standardisation organization CEN. The first standard produced by WG1 is ISO 14258, *Concepts and rules for enterprise models* (ISO, 1998a). This is a high-level standard defining the nature of enterprise models with the vision that compliant models could be used to design, analyse, and eventually, operate enterprises. The rules for models are based on classic systems theory, with the assumption that an enterprise or groups of processes is basically a system and that it can be designed and analysed as such. ISO 14258 is the most general standard of the planned series from WG1. A second standard has been developed: ISO 15704, *Requirements for enterprise-reference architectures and methodologies* (ISO, 1998b). ISO 15704 defines the requirements that enterprise-reference architectures and methodologies must have to be considered complete. This will be useful to those trying to improve an enterprise infrastructure or its processes, and who will create an enterprise architecture of their own that is specific to a company, industry, or purpose. This standard will help guide that creation process. Previous work in CEN had developed ENV 40003 Framework for Enterprise Modelling, which is a partial implementation of these requirements (ENV40003, 1990). These enterprise-reference architectures and methodologies will help carry out all types of enterprise-creation projects as well as any incremental change projects required by the enterprise throughout the whole life of the enterprise including enterprise creation, major enterprise restructuring efforts, and incremental changes affecting only parts of the enterprise-life cycle. The necessity for modelling and integrating the enterprise is in international standards, and ISO publishes these standards under following numbers and definitions (EA_Standards, 1999) in summary in accordance with Figure 2.5:

- preEN/ISO 19439 : Enterprise Integration - Framework for Enterprise Modelling, ISO TC 184/SC5/WG1 - CEN TC 310/WG1, 2003
- preEN/ISO 19440: Enterprise Integration - Constructs for Enterprise Modelling, ISO TC 184/SC5/WG1 - CEN TC 310/WG1, 2003
- ISA 95.00.01: Enterprise-Control System Integration , IEC/ISO JWG15, 2002
- ENV 13550 : Advanced Manufacturing Technology - Systems Architecture - Enterprise Model Execution and Integration Services, CEN/TC310, 1999
- IS 15704: Requirements for Enterprise Reference Architecture and Methodologies, ISO TC 184/SC5/WG1, 1998
- IS 14258: Industrial Automation Systems - Concepts and Rules for Enterprise Models, ISO TC 184/SC5/WG1, 1998

- ENV 12204: Advanced Manufacturing Technology - Systems Architecture - Constructs for Enterprise Modelling, CEN TC 310/WG1, 1996
- ENV 40003: Computer Integrated Manufacturing - Systems Architecture - Framework for Enterprise Modelling, CEN/CENELEC, 1991

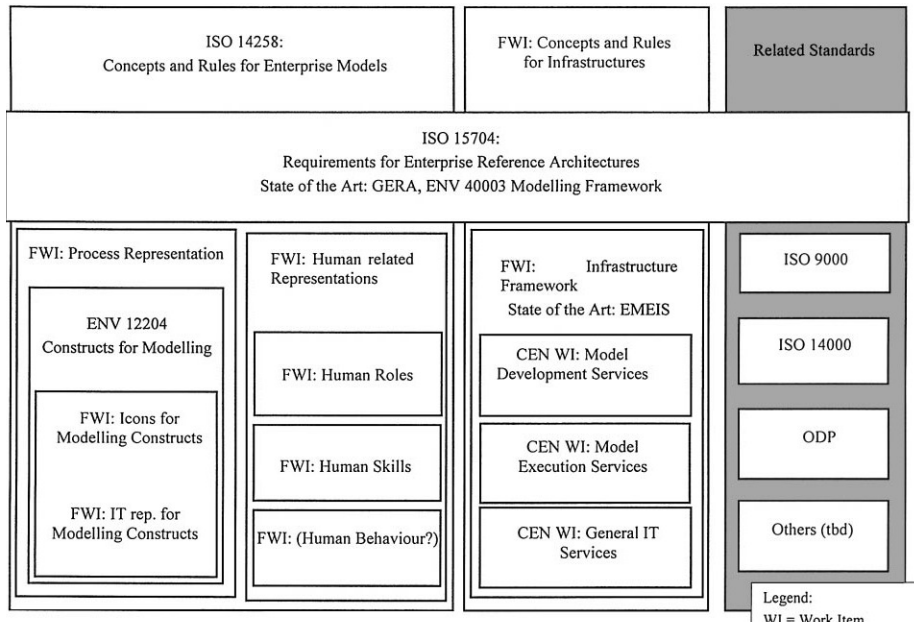


Figure 2.5 Available standards in enterprise modelling.
Reference: (Kosanke & Nell, 1999)

ERP, CRM and ISO 9000 applications should be in parallel with the context of the enterprise model and consider exactly the same objectives. Enterprise modelling schemes include not only classes of charts and flows, but also ontology for the entire enterprise. Especially the ontology for quality activities is in accordance with the definitions in ISO documents. Therefore, enterprise modelling provides a significant infrastructure for the further applications and certifications for the enterprise. The hope at this point is that the Turkish firms are aware of these standards as soon as possible.

Future standardisation work in enterprise engineering has to focus on the needs of electronic commerce and to support inter-operability in extended and virtual enterprises. This new paradigm in enterprises will flourish only if partners can exploit market opportunities on short notice and can establish their enterprise fast enough to take advantage of the opportunity. The ICEIMT'97 -International Conference on Enterprise Integration and Modelling Technology, during the five workshops and conference, suggested further research, development and standards work (Kosanke & Nell, 1997). Additional items have been identified in the course of the standardisation work itself. Their purpose is to help ISO TC184 SC5 WG1 to plan and prioritise new standardisation work in the enterprise-representation domain that is needed by industry. Categories for projects are: process representation, human role representation, integrating infrastructure, terminology facility, and standards landscape.

2.10 Enterprise Knowledge Development

An enterprise model is the meta-model of all application systems to be implemented within an enterprise, i.e., information architectures and knowledge models, and enterprise resource management systems. Among these application systems, knowledge development is the most complex process, and the more efficient the enterprise model, the more successful is the knowledge model.

Enterprise knowledge development (EKD) is performed by responsible agents having the freedom to decide how to proceed according to their evaluation of their situation. Agents do not necessarily follow a predefined plan of action. Defining and implementing change requires a number of decisions to be made: what to consider in the existing organization; what should be improved; the alternative solutions; and the selection of the most appropriate solution. The EKD process cannot be ad-hoc and chaotic. It cannot be only based on intuition and personnel behaviour of engineers and stakeholders (Rolland, Nurcan, & Grosz, 1999). Thus, enterprise knowledge model needs for another meta-model which is provided by an enterprise model.

When enterprises are taken into account from the social and behavioural viewpoint, one can see that many conditions and situations can only be described with qualitative analysis and conceptual models. In order to deal with enterprise knowledge complexity, a multi-perspective approach is advocated. The key aspects of this approach are encapsulated in Figure 2.6. The task of enterprise knowledge modelling is viewed as a co-operative activity which exploits the contribution of different modelling views, each encompassing a specific type of knowledge. When combined, these perspectives will produce an integrated, consistent and complete knowledge model of the enterprise analysed. Within this multi-perspective approach enterprise analysis is based on two mechanisms: reasoning within a perspective; and reasoning across different perspectives in order to allow each individual step in the analysis process to exploit the most appropriate knowledge source. As can be seen in Figure 2.6, knowledge regarding enterprises can be logically partitioned into three categories (or views): (a) the ‘Goals’ view, i.e., the enterprise objectives and the ways that these may be realised; (b) the ‘Operation’ view, i.e., the enterprise structures and functioning that realise the objectives; and (c) the ‘Rationale’ view, i.e., justification, explanations and arguments supporting the different objectives and corresponding designs of the operations.

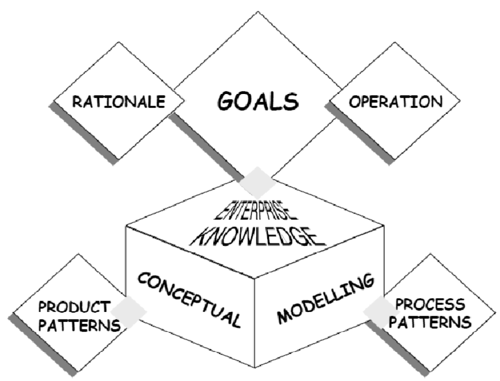


Figure 2.6 Enterprise modelling framework.
Reference: (Loucopoulos & Kavakli, 1999)

In accordance with the concept of enterprise modelling and knowledge management issues, this dissertation proposes an analytical way to integrate enterprise modelling characteristics and views with enterprise goals, processes and operations based on Quality Function Deployment in chapter 6.

The major and the most difficult component of enterprise modelling is the information model. For this component, system or software development tools are commonly used. This difficulty comes from the way the works are handled within the enterprise. Generally, the original design documents of the system may probably be lost or out of date; only a few business processes may be documented, only technical and procedural information is available but not the rationale of their design, i.e., why the system was designed that way; very few exception-handling procedures may be correctly documented, this knowledge remains in the head of operators; each individual involved in the manufacturing system has a different perception to his colleagues of his role or about the system operation; and finally the system never works the way it was planned (Vernadat, 1996).

General models of software development in enterprise modelling focus on analyzing data flows and transformation. This kind of modelling only accounts for organizational data and also for the related portion of the process in interaction with the data. The correct integration of information systems in the business administration requires, however, a more integrated approach to system specification (Snoeck, Agarwal, & Basu, 1998). The re-framed Zachman framework for information systems architecture (Zachman, 1987) proposes a layered approach to the specification of an information system that puts information systems in a much larger context (Snoeck, 1999). Most current software development methods have no distinction between business and information functionality. They typically group in a business object not only the core business attributes and business routines, but also input and output procedures. Some methods offer somewhat analogous concepts. With some object oriented software engineering inherited concepts in UML allows also to distinguish entity objects, as opposed to interface and control objects. In these methods however, the choice of techniques is critical in addressing the appropriate functionality level: the use of flows or streamed communication mechanisms (such as message passing) may contain implicit implementation choices, which should be addressed in the implementation, and not in the specification (Jacobson, 1992, p.51) .

Many manufacturing companies are taking advantage of recent advances in software and hardware technology to install integrated information systems called Enterprise Resource Planning (ERP) packages. These systems can provide seamless and real time data to all who need it. Essential steps in the implementation of an

ERP-package are the development of an enterprise model, business process re-engineering and the identification of desired information system services. The development of an enterprise model allows to gain better insight in the business functioning. The enterprise model centralizes all the business rules that remain valid even if there is no supporting information system. In the context of the evaluation and acquisition of an ERP package, business modelling allows to match the own business rules against the business rules supported by the ERP package. The more of the own business rules are supported by an ERP package, the less changes in business functioning will be required when implementing that package. In addition, the separation of business rules from functionality requirements allows for a better insight in the cost of requested changes, e.g. changes to the enterprise model equal to changes in the basic business rules. Hence, these kinds of changes will in general be more costly than changes to services. Indeed, a change in the enterprise model will generally require changes to all services that are based on the modified portion of the enterprise model. A full comparison of the own enterprise model and the enterprise model supported by the ERP package allows to better evaluate which part of the business will have to be adapted to the use of this package. As the modification of business rules implies a more fundamental change in business functioning than a modification in tasks and workflow, the availability of a business model is an interesting tool in any ERP evaluation process (Snoeck, Agarwal, & Basu, 1998).

2.11 Current Trends Of Requirements Analysis In Enterprise Modelling: Unified Enterprise Modelling (UEML) Project

The UEML project (which is an IST Thematic Network funded by the European Commission in the Sixth Framework Program) was set up in an attempt to contribute to the solving of the problems of multiple Enterprise Modelling Languages (UEML, 2001). The long term objective of UEML is the definition of a Unified Enterprise Modelling Language, which would serve as an inter-lingua between enterprise modelling tools. UEML language is supposed to provide a flexible modelling platform to support enterprise engineers during the modelling process. In details UEML is a platform that

- Provides the business community with a common visual, template based language to be used on top of the most commercial enterprise modelling and workflow software tools; Provide standardised mechanisms for sharing knowledge models and exchanging enterprise models among projects, and overcoming tool dependencies;

- Supports the implementation of open and evolutionary enterprise model repositories to leverage enterprise knowledge engineering services and capabilitiesin order to prepare this long term objective.

The UEML project was initiated with the objective to create and manage a working group aiming:

1. To Create a European Consensus on a common Enterprise Modelling Language and to facilitate interoperability in the frame of on-going standardisation efforts in this domain. The common language representing this consensus will be defined in terms of a core set of modelling constructs.

2. To build a UEML demonstrator portal with services and contents to support and promote, test, enable industrial validation, and to collect comments on the proposed Modelling Language Constructs.

3. To prepare the launching of a project to define, implement, extend, adapt, manage, and re-configure the various constructs of language-variants as will be implemented by industries and business projects.

From a technical point of view, it is therefore necessary either to provide peer-to-peer gateways between the proprietary languages and models or to use a *common format* (like a UEML) for exchanging these models (which are embedded in distinct tools and represented in proprietary formats). None of the tools for enterprise modelling studied in this state of the art provide such a common format.

Currently, several Meta-Modelling Languages (and also tools) exist but none of them are specifically targeted for the definition of enterprise modelling languages and enterprise engineering methodologies. The reason is that these meta-modelling languages were often developed to design and implement *information systems, knowledge-base systems and computer-based infrastructures* (environments) allowing to program meta-models. They were not developed with the specific objective to support the definition of enterprise modelling languages. A *UEML could be defined as a content-dependent domain-specific meta-model* through a content-independent meta-model. The UEML might just use content independent meta-modelling techniques as a *way for its definition*. However, the notion of meta-modelling technique is *relative* (UEML, 2001).

This analysis of the state of the art demonstrates the need to define and develop a UEML approach to solve the current problems faced by enterprise modelling domain. But such UEML approach can only be successful and effective at two conditions:

- That it provides a global approach of interoperability among enterprise modelling software going further than just providing a common format of exchange;

- That it makes clear and effective the link between the effort of enterprise modelling and enterprise applications and software (See Figure 2.7).

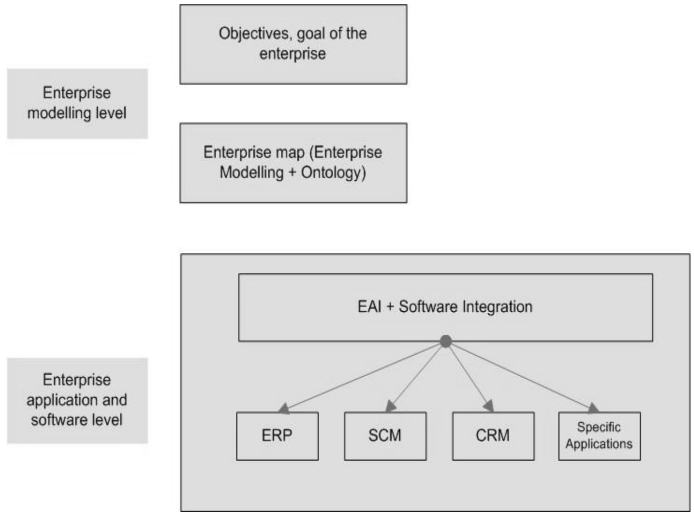


Figure 2.7 The distinction and link between the enterprise modelling and enterprise architecture layers.
Reference: (UEML, 2001) / (Interop Project, 2003)

As stated earlier, a common exchange format, if deemed successful, cannot be described independently of mappings to and from existing EMLs. Furthermore, this requires the explicit definition of meta-models of the involved languages and of the mapping among their concepts. However, in order to avoid that UEML as a common format becomes yet another language among the large set of existing ones, it requires a larger view of interoperability among enterprise modelling tools. The UEML language and approach must be flexible to be able to cope with future proprietary emerging languages and with the evolution of UEML itself. The long term objectives of a UEML approach would then be to provide the necessary concepts and tools to achieve the following:

- *Interoperability* between already existing supporting *tools as well as newly developed tools*,
- Well-founded integration base between distinct *enterprise modelling languages*,
- Consistent global models on which also distinct *methodologies* can be integrated,
- Improvement of *existing methodologies* and definition of *new methodologies.*

These objectives pose a number of requirements on the UEML approach:

- The availability of concepts, methods and tools to properly define enterprise modelling languages (existing ones, new emerging ones, UEML, its extensions and particularisations for specific purposes or applications);
- The availability of concepts, methods and tools to properly define relations existing among distinct enterprise modelling and a UEML and relations existing between models created with different enterprise modelling languages and UEML;

The specification of an *open architecture* in which all these things can be implemented to provide an evolutionary multi-language platform for enterprise modelling centered on UEML. This platform would allow creating *coherent, global and logically centralised* (integrated) models of the enterprise but which may be distributed within different enterprise modelling applications at a *physical level*. Additionally, this platform would allow a seamless integration and use of the specific functionalities available in enterprise modelling tools. In the interoperability of enterprise modelling languages project, not all of the suggested applications and objectives of a UEML can be achieved. However, from the analysis provided in this conclusion section, it seems reasonable to first tackle the problem of integrating distinct modelling languages. This approach is useful not only for making possible the integration between tools supporting distinct modelling languages but also to investigate the feasibility of such a UEML approach (by applying a *systematic methodology* to achieve it) and to show some benefits of the UEML from a methodological point of view (UEML, 2001).

Whenever an enterprise takes part in a network of enterprises, the number of coexisting EMLs is likely to increase. Therefore, *translations* between couples of languages are also called *peer-to-peer translation* (bidirectional arrows in Figure 2.7). These translations are possible, though difficult and costly to carry out, and not suitable within networks of enterprises where fast changes of partners are usual. The other disadvantage of peer to- peer translations is related to the “maintenance of a global consistency” due to a clear lack of *global, unique and consistent vision* about the knowledge shared between the enterprises (each modelling language does not cover all enterprise aspects as it is shown in Figure 2.8 where three modelling techniques, e.g. MOOGO, eMAGIM and METIS; embrace respectively resource monitoring, decision support and enterprise planning).

Therefore, to reduce dramatically the number of *interfaces* (peer-to-peer translations) needed to communicate a set of enterprises, and to increase the achievement of a "global consistency", it is really useful to define an *intermediate federator language* which eventually allows to represent a *unique, consistent and modular vision* on the shared (or integrated) knowledge of the whole set of enterprises. Thus, such a language, generally called UEML, does not substitute existing modelling languages, as its target is to provide effective support for enterprise model translation and integration. In this sense, a UEML should be equipped with *standard translation mechanisms* to and from existing modelling languages. Furthermore, the UEML permits enterprises to retain their enterprise modelling languages without forcing them to use the UEML itself (Berio, Anaya, & Ortiz, 2004). Figure 2.8 shows this integration:

UEML 1.0 structure is feasibility study of interoperability of common enterprise modelling languages. Nowadays the Project group studies on specifically interoperability of enterprise modelling languages in the Project with the name "interop" which is being published in (Interop Project, 2003) as the second version of UEML(UEML 2.0).

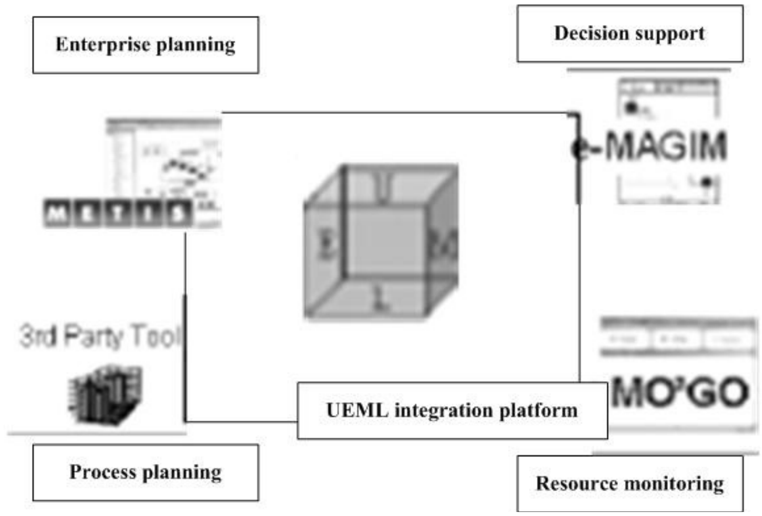


Figure 2.8 Integration in UEML.
Reference: (Berio, Anaya, & Ortiz, 2004)

The interest of this approach is to start from the users requirements to develop UEML and not to use a traditional empirical approach. These interests gather the expertise of experts from several domains and in particular from organization and information technology, which are traditionally opposed (Ducq, Chen, & Vallespir, 2004).

The project group first determines the user requirements of UEML with the process given in Figure 2.9 and 2.10:

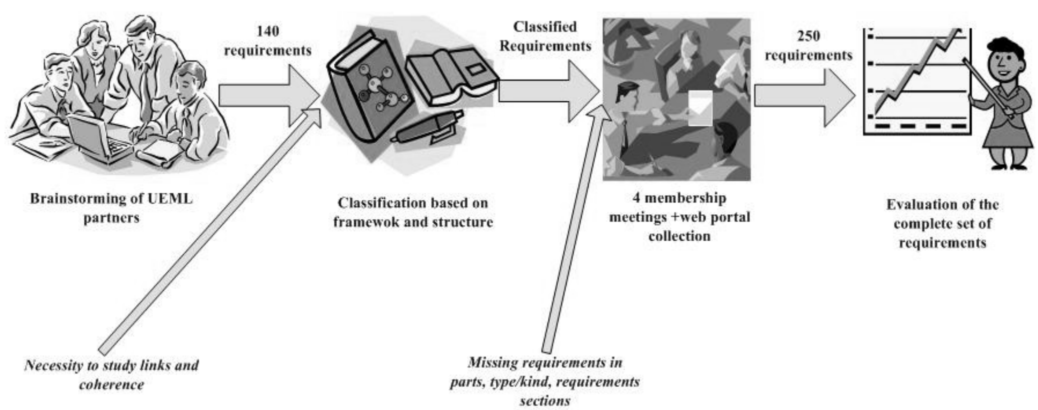


Figure 2.9 UEML structure.
Reference: (Berio, Anaya, & Ortiz, 2004)

The major purpose to present Figure 2.9 and Figure 2.10 is to show the requirement analysis part of a project to compare with the context of this thesis. There exist common phases such as starting with goal statement with top down approach.

As the collection of the requirements is represented in Figure 2.10, these approaches just classify the requirement to understand them. Beside the classification, the proposed approach can calculate the weights and importance of each requirement in the enterprise. Furthermore, the importance of each item in the upper level can be deployed to the lower level by using QFD matrices.

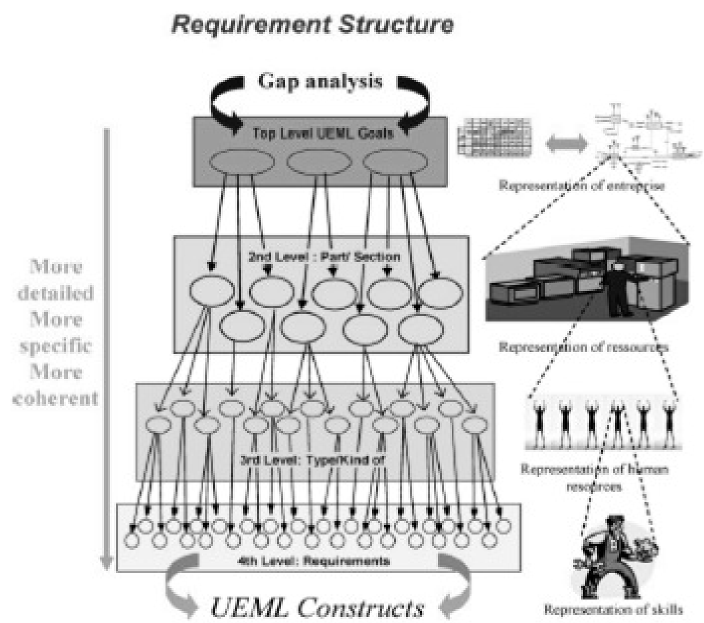


Figure 2.10 Decomposition of UEML user requirements.
Reference: (Ducq, Chen, & Vallespir, 2004)

Consequently, the improvements in UEML project continuing as Interop Project can contribute to the flow of this thesis from two perspectives. The requirement analysis phase can be compared with the proposed approach, and also this new trend can provide a new viewpoint for the future phases.

CHAPTER THREE
ENTERPRISE REFERENCE ARCHITECTURES AND FRAMEWORKS

3.1 Definitions

"Reference architectures are intellectual paradigms which facilitate analysis and accurate discussion and specification of a given area of discourse. They provide a way of viewing, conceiving, and talking about an issue" (Vernadat, 1996).

An enterprise architecture focuses on modelling different domains relevant for businesses or organizations. A major issue is how to express and maintain the relations between different modelling domains. Current architectural support focuses mainly on modelling techniques and language for single domains. For enterprise architectures it is important to have the flexibility to create cross domain models and views in which inter-relations are made explicit. Therefore, a language for enterprise architecture models should pay particular attention to the relations between domain models. Buuren, Jonkers, Iacob, & Strating (2004) presents a general approach to derive an operator that allows for the composition of relations in architecture description languages. This general approach opens the door for a number of interesting application areas, two of which are worked out in more detail: the creation of more modelling flexibility, by allowing leaving out certain details, and automated abstraction and complexity reduction of models facilitating stakeholder-specific visualizations. For a specific enterprise architecture modelling language, it explicitly derives this composition operator.

To carry out the project of master planning and implementation of an "integrated enterprise system" is an extremely complex process which involves different technological, human and organizational elements. In order to make the study of existing systems and the design of new and more advanced systems easier by reducing the complexity level, it is necessary to establish a step by step development methodology and to formalize the creative process in each phase of the whole project (Pantakar, 1995).

The architecture must guide the development and application of all of the disciplines involved in the enterprise integration project, systematically modelling all parts of the life cycle of the enterprise. This means the states of definition, specification, detailed design, physical implementation or construction and maintenance, till its obsolescence. All the activities in the enterprise integration project must have their place in the reference architecture and the enterprise development program must be detailed step by step (Chalmeta, Campos, & Grangel, 2001). All these issues bring the idea that there should be a specific job position and definition which will intensively handle and manage the enterprise model with an appropriate architecture. This necessity generates the concept of "enterprise engineering".

An enterprise can be viewed as a complex 'system' with multiple domains that may influence each other. In general, architectures are used to describe components, relations and underlying design principles of a system (Society, 2000). Constructing architectures for an enterprise may help to increase insight and overview required to successfully align the business. Although the value of architecture has been recognised by many organizations, mostly separate architectures are constructed for various organizational domains, such as business processes, applications, information and technical infrastructure. The relations between these architectures often remain unspecified or implicit. In contrast to architectural approaches for models within a domain (e.g., the Unified Modelling Language, UML (Booch, Rumbaugh, & Jacobson, 1999) for modelling applications or the technical infrastructure or the Business Process Modelling Notation (BPMN, 2003) for modelling business processes), enterprise architecture focuses on establishing a coherent view of an enterprise. The term refers to a description of all the relevant elements that make up an enterprise and how those elements inter-relate. Models play an important role in all approaches to enterprise architecture. Models are well suited to express the inter-relations among the different elements of an enterprise and, especially if they can be visualised in different ways, they can help to alleviate the language barriers between the domains (Buuren, Jonkers, Iacob, & Strating, 2004).

Taking the basis of UML, another modelling platform is developed especially for business modelling called Business Modelling Language (BML) (Wahlander, Nilsson, & Skoog, 1998). There is a growing need for Enterprise Application Integration (EAI) technologies, which align the applications of an organization to its business processes. Such technologies require an adequate methodological support so that well-structured and easily understandable models can be constructed. BML is a communication oriented process language, which means that it focuses on describing interactions between systems through the sending and receiving of messages. This makes the language suitable for application integration. Another important advantage of BML is that the language can be used for business specification and design as well as in the execution of systems. This means that the same language can be used in different phases of a system's life cycle: in feasibility analysis, in requirement specification, in the design and implementation phases, and even in the operation phase. This enables different categories of stakeholders to use the same language for different purposes. The language can also be used directly as an implementation language and to some extent replaces ordinary programming languages. Another advantage of using BML is that it is possible to describe and partition the interaction and interfaces between processes that work concurrently. Concurrency is common in application integration, e.g. when several applications are to be updated in parallel. The possibility of partitioning in BML reduces the complexity of handling large systems, through creating manageable and understandable parts with limited dependencies (Johannesson & Perjons, 2001).

A similar movement towards integrated models can be recognized in the Model Driven Architecture (MDA) approach to software development (Frankel, 2003). MDA is a collection of standards of the Object Management Group (OMG) that raise the level of abstraction at which software solutions are specified. Typically, MDA results in software development tools that support specification of software in UML instead of in a programming language like Java. Recently, OMG has extended its focus to more business-oriented concepts and languages, to be developed within the MDA framework. These developments make MDA just as relevant for enterprise architecture as it is now for software development. The MDA trend reflects the growing awareness that it is important to take into account business considerations in software development decisions. Therefore, enterprise architectures form a natural starting point for automated software engineering. Figure 3.1 shows the meta-model of a general structure of enterprise architectures.

This meta-model takes its sources from the unified modelling language and represents a model driven approach for each enterprise architecture model. UML structure is also discussed in chapter 4 in details.

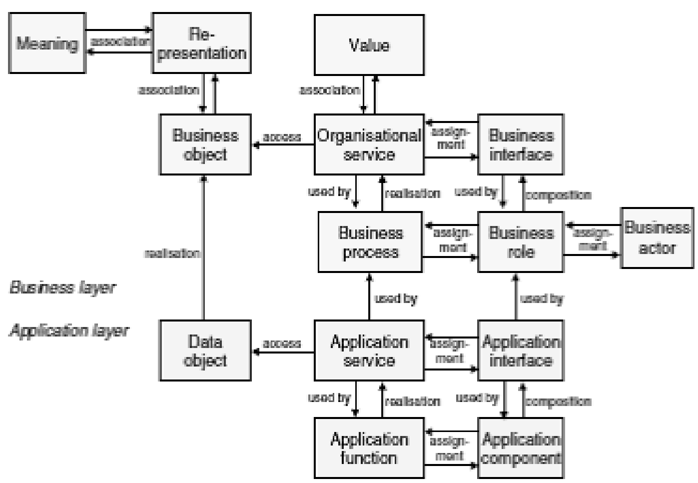


Figure 3.1 Metamodel of the core of the enterprise architecture description language.
Reference: (Buuren, Jonkers, Iacob, & Strating, 2004, p.45)

A key function of reference architecture for enterprise creation, operation, and analysis is to determine, in specific and generic ways what characteristics of an enterprise are necessary to analyse to help achieve an improved degree of enterprise integration. Once the key elements of these characteristics are logically arranged into a reference architecture, there exists an excellent reference architecture for an enterprise model. Therefore, one could view the enterprise-reference architecture as a high level enterprise model or a meta-model for a set of enterprise models. The elements of the reference architecture would be a framework that would indicate the key things in the enterprise that one should consider when creating, analysing, or using an enterprise model (Bernus, Nemes, & Williams, 1995); (Vernadat, 1996).

The following sections first present the enterprise engineering concept, then the most representative architectures internationally applied in the scientific researches and implementations in practice, i.e., ISO work, CEN-ENV40 003, CIMOSA, GRAI/GIM, PERA, ARIS, GERAM, and related ones.

3.2 ISO TC184/SC5/WG1 (ISO WORK)

To understand potential areas subject to standards development in manufacturing systems, Sub-Committee 5 (SC5) of ISO TC 184 has produced a Reference Model for shop floor production standards documented in the ISO Technical Report 10314. The aims of the Reference Model are (ISO, 1990):

- to provide a conceptual framework for understanding discrete parts manufacturing; and
- to be used to identify areas of standards necessary to integrate manufacturing systems.

The ISO Reference Model described in Part 1 of Technical Report 10314 is structured into three sub-models:

1. A **context for shop floor production,** which identifies major functions (e.g. finance, sales order system, materials resources planning, engineering/CAD, production, and finished goods storage) of discrete parts manufacturing and major information flows among them, the shop floor model is based on National Institute of Standards and Technology (NIST), but there are some differences. While the NISTS model (see Figure 3.2)has five levels (facility, shop, cell, work station, and equipment), the shop floor production model (see Table 3.1) restricts itself to the four lower levels (section/area, cell, station, and equipment)
2. The **shop floor production model (SFPM),** which represents a four-level hierarchy of generic shop floor production activities (see Table 3.1).

3. The **generic activity model (GAM),** which depicts activities and flows (materials, information, and resources) between activities. The purpose of the generic activity model (GAM) (not a standard) is to provide a way to generically describe the activities found at each level of the shop floor production model. It is based on a graphical representation. It is sufficiently general to represent any shop floor production activity in terms of its inputs and outputs (referred to as subjects) and its actions (see Figure 3.3). Combining actions, subjects, and levels with the values indicated above gives several matrix representations for identification procedures of standards. Horizontal and vertical interactions between levels of the Shop Floor Production model can be analyzed.

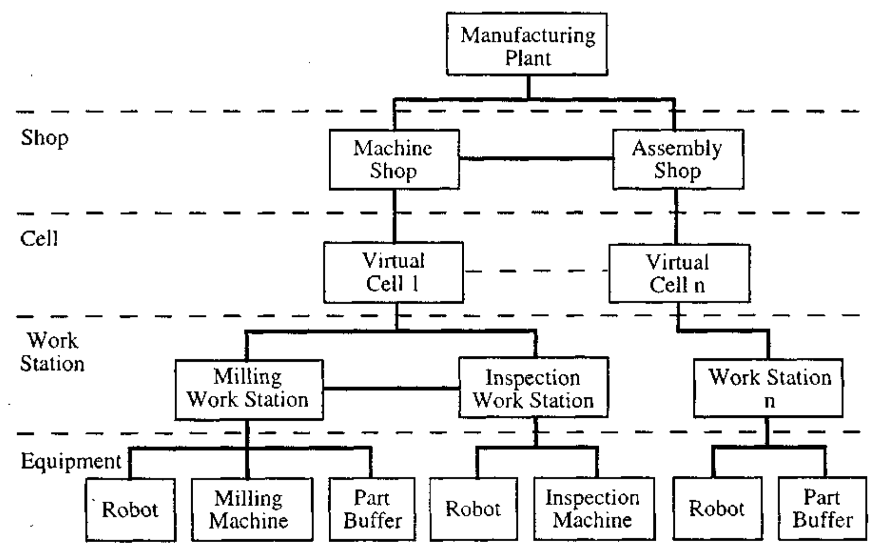


Figure 3.2. NIST model for manufacturing plants.
Reference: (Vernadat, 1996, p.34)

Table 3.1 Shop Floor Production model in ISO TC184/SC5/WG1

	Level	Sub-activity	**Responsibility**
4	Section Area	Supervise shop floor production process	Supervising and coordinating the production and supporting the jobs and obtaining and allocating resources to the jobs
3	Cell	Coordinate shop floor production process	Sequencing and supervising the jobs at the shop floor production
2	Station	Command shop floor production process	Directing and coordinating the shop floor production process
1	Equipment	Execute shop floor production process	Executing the job of shop floor production according to commands

Reference: (Vernadat, 1996, p.34)

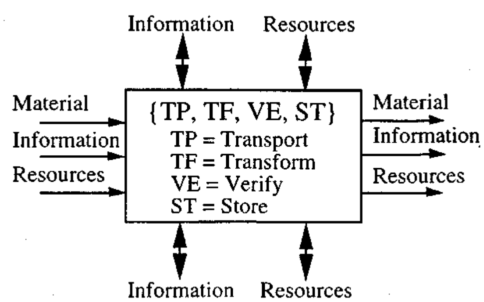


Figure 3.3 The generic activity model in ISO TC184/SC5/WG1.
Reference: (Vernadat, 1996, p.35)

In 1990, ISO TC184/SC5/WG1 has started work on a 'Framework for Enterprise Modelling'. The goal of this working group on integrated systems architecture is to establish a framework to coordinate existing, emerging, and future standards for the modelling of manufacturing enterprises in order to facilitate computer-integrated manufacturing (CIM). The work is based on analysis of previous proposals as described in the next sections. It covers such items as terminology for enterprise modelling, scope of enterprise modelling, modelling concepts, process of enterprise modelling, and applications to CIM (Vernadat, 1996).

3.3 CEN ENV 40 003

The European Pre-Standard ENV 40 003 entitled 'Framework for Enterprise Modelling' provides a framework for future standardization activities in the area of computer integrated manufacturing enterprise modelling (CEN, 1990). Its goal is to help in the identification and positioning of necessary standards in the area of CIM, and to define a framework for computer-based modelling of enterprises, focusing on discrete parts manufacturing (Vernadat, 1996, p.37).

The ENV 40 003 has been prepared by the European Committee for Standardization (CEN/CENELEC AMT/WG-ARC Working Group on CIM Systems Architecture). It is now under the responsibility of CEN Technical Committee TC310 Working Group 1. The objective of the working group is 'to ensure that the requirements of European industry are met, so that maximum advantage can be taken of standardization for enterprise modelling and the use of development environments that will influence the industrial organization, management, and manufacturing approach to improve efficiency. The framework has been developed from a substantial contribution from ESPRIT projects, and especially AMICE and its CIMOSA architecture, with further inputs from industry and academia (CEN, 1994).

Like the ISO Work, the relation, the ENV 40 003 defines different layers to guide the structuring and development of future standards for enterprise modelling.

It is structured according to three dimensions (Figure 3.4):

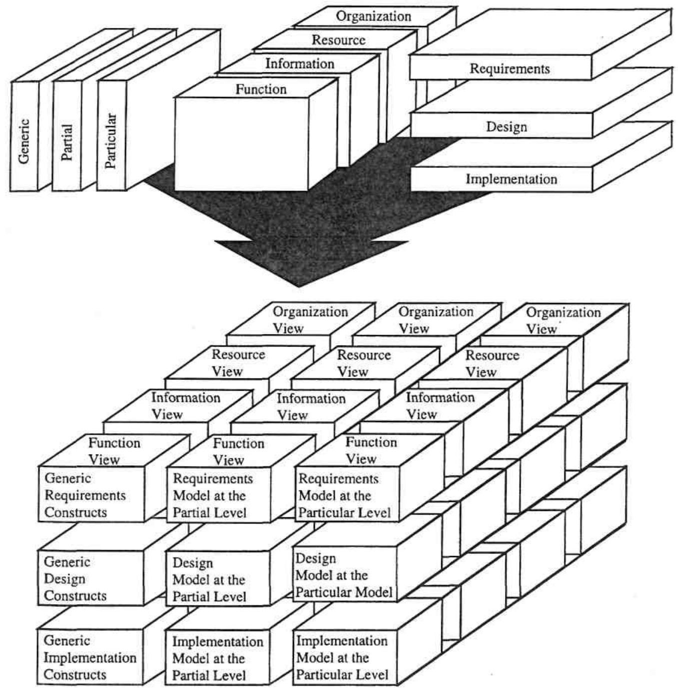


Figure 3.4 CEN ENV 40003 framework for enterprise modelling.
Reference: (CEN, 1990)

- **Dimension of Genericity**
  - the generic level, which defines the basic modelling constructs for components, constraints, rules, terms, services, functions, and protocols;
  - the partial level, which contains partial models;
  - the particular level, which describes enterprise specific knowledge using constructs of the generic level.

- **Dimension of models**
  - requirements models, which define enterprise operations to be done (and possibly how they could be done) in a business sense and terminology, in terms of enterprise operations, information, resource requirements, responsibilities, and authorities without any reference to implementation options or decision;
  - design models, which specify how the enterprise operations are to be performed, that is, the actions and processes that are to be

performed, the information entities, resources, and organizational structures to be used to achieve the enterprise requirements;
    - implementation models, which describe the means and/or rules to be used in executing the enterprise operations as defined in the requirements models.

- **Dimension of views**
    - the function view, which provides a hierarchically structured description of the functions, behaviour (dynamics), and functional structure (statics) of the enterprise with relevant inputs and outputs;
    - the information view, which provides the description of a structured set of enterprise objects that were identified in the other views;
    - the resource view, which provides a description of the resource organization of the enterprise, i.e., the set of resources required to execute the enterprise operations;
    - the organization view, which provides the description of the organizational structure of the enterprise, the responsibilities of the individuals, and the organizational units within the enterprise.

CEN and related architectures provide the general framework for computerized production systems, and there is no invariable property and the necessary tools or characteristics can be added according to the type of production system. CEN ENV 40003 is enlarged and improved, architectures and frameworks are derived (Shorter, 1999).

3.4 CIMOSA (Computer Integrated Manufacturing Open System Architecture)

CIMOSA, the European Open Systems Architecture for CIM, has been developed by the AMICE Consortium as a series of ESPRIT Projects jointly financed by the European Commission and project partners (30 companies in total) grouping CIM suppliers, large users, and academia from 1986 until 1994. Other ESPRIT projects have also contributed to CIMOSA by testing and validating CIMOSA principles

(AMICE, 1993); (Zelm, Vernadat, & Kosanke, 1995). The complete technical description of CIMOSA is documented in the CIMOSA Formal Reference Base published and maintained by the (CIMOSA, 1996).

The goal of CIMOSA is to help companies to manage change and integrate their facilities and operations to face worldwide competition and to compete on price, quality, and delivery time. The basis to achieve this is an integrated enterprise model. CIMOSA provides a consistent architectural framework for both enterprise modelling and enterprise integration as required by CIM environments, which comprises (Vernadat, 1996, p.41):

- a general definition of the scope and nature of CIM;
- guidelines for implementation;
- a description of constituent systems and subsystems;
- a modular framework complying with international standards.

The CIMOSA modelling framework is based on three orthogonal principles:

1. The **derivation principle,** which advocates to model enterprises according to three successive modelling levels (iterations among these levels are of course allowed):
   (a) requirements definition to express business needs as perceived by users;
   (b) design specification to build a formal, conceptual, and executable model of the enterprise system (time is considered);
   (c) implementation description to document implementation details, installed resources, exception handling mechanisms, and taking into account the system non-determinism.
2. The **instantiation principle** based on three generic layers:
   (a) a generic layer containing generic building blocks and building block types (structured as taxonomies) as the elements of the modelling language (or modelling language constructs) to express any model (partial or particular);
   (b) a partial layer containing libraries of partial models classified by industry sectors to be copied and used in particular models; and

(c) a particular layer containing particular models, i.e., company specific models of parts of a given enterprise.

3. The **generation principle,** which recommends to model manufacturing enterprises according to four basic but complementary viewpoints (other views could be defined):

CIMOSA components construct a cubic framework including four major views on one side as given in Figure 3.5.

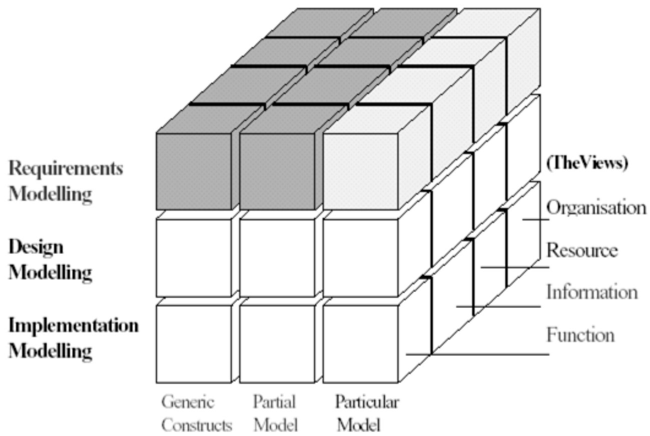


Figure 3.5 CIMOSA modelling framework.
Reference: (CIMOSA, 1999).

The CIMOSA cube, as shown in Figure 3.5, consists of two main parts; the reference architecture and particular architecture. The particular architecture is a set of models documenting the CIM environment of the business users in the process of building their own particular architecture. CIMOSA cube establishes three modelling levels; requirements definition, design specification, and implementation.

One of the other sides of this cube is developed based on three generic levels; generic, partial, and particular layer. These layers indicate the level of detail included in the models. The generic layer acts as a library of basic building blocks for constraints, rules, functions, and protocols. The partial layer has predefined partial models which can be applied to particular models. Particular layer is a model of a specific enterprise which is built from basic blocks and partial models.

The last side of the CIMOSA cube represents the modelling views; function, information, resource, and organization. These views are used for modelling manufacturing enterprises.

Function view models the enterprise structure, functionality, control, and behaviour in terms of domains, domain processes, enterprise activities and business processes. This view is developed in three stages. The first stage gives a macro definition of the total enterprise in order to identify what has to be done, which is made possible by means of the business processes. In the second stage, the business processes are defined by means of events, results, and other related sub-processes. In the final stage, the functionality is defined by identifying the inputs and outputs of the processes (Presley, 1997).

The functional part represents the static part and is composed of (Tham, 2000):

- *Objectives and constraints* limiting the definition and specification of the enterprise function,
- *Functional description* which describes the action required to produce the required output from the inputs provided,
- *Required capabilities* consist of minimum requirements on the descriptive attributes of the function
- *Inputs and outputs* describe the objects that the function needs for its execution and that it produces as a result of the execution.

The behavioural part forms the dynamic section of the enterprise section and includes (Tham, 2000):

- *Objectives and constraints*- define only those objectives and constraints that are applicable to the execution of the domain/business processes,
- *Set of procedural rules*- defines the desired sequence of the enterprise functions in the form of a flow of control. This is the essential part of the behaviour of the enterprise function. Each procedural rule consists of a

sequence number, name of enterprise function (domain process, business process or enterprise activity), a list of ending statuses, and actions to be taken for each status.

- *Enterprise events*- initiate the execution of domain and/or business processes by activating the processing of appropriate procedural rules,
- *Ending status*- a list of ending statuses of the processes required for further processing.

The structural part of the enterprise is composed of (Tham, 2000):

- *Where used component*- a list of domain process or business process where the enterprise function is used,
- *Comprises* – a list that identifies the enterprise functions in the next lower level of decomposition of a given enterprise function.

Information view represents enterprise objects, object views, and information elements. The enterprise object represents real world entities of the enterprise while object view represents the state of the enterprise objects. The information element is any piece of information or data. This view includes four types of information (Presley, 1997):

- Product- information about the products and production processes,
- Manufacturing planning and control- information related to the handling of orders,
- Shop-floor- information about the manufacturing operations,
- Basic information- supports many functions or departments such as company standards and guidelines.

The resource view contains all of the relevant information on enterprise resources- machines/equipment, people and application programs. These resources are classified as active resources which are capable of performing operations (e.g.

machines) and passive resources which are incapable of performing any operations (e.g. tools) (Presley, 1997).

The organizational model generated in **the organization view** consists of all of the relevant information on the responsibilities within the enterprise and allows gathering and structuring the different responsibilities for functions, information, and resources in the enterprise. This view is built upon the following constructs: organizational units, organizational cells, responsibility, and authority. Organizational units are the lowest level in this view and organizational cells are collections of organization units that describe an organizational area of the organizational structure. Responsibility is an ability provided to an organization unit to make decisions about a given area of competency. Authority is an ability provided to an organization unit to make decisions about other organization units (Presley, 1997).

Other layers of the CIMOSA cube are similar to the CEN framework mentioned in section 3.2.

CIMOSA provides a process oriented modelling concept that captures both the process functionality and the process behaviour (Figure 3.6). It supports evolutionary enterprise modelling, e.g., the modelling of individual enterprise domains (DM) which may contain one or several individual processes. Domains and processes are defined by the user according to his/her needs for controlling the business operations. Processes themselves should be defined as significantly large pieces of functionality which produce a certain end-result for a defined customer. Customers may be internal or external to the enterprise. CIMOSA always models the relations to the internal and external environment. This allows models to be integrated with other process models at a later point in time. The relations will become the links to the added models. To handle complexity, CIMOSA follows an enterprise engineering concept which separates functionality (EA: Enterprise Activity) and behaviour (BRS: Behavioural Rule Set) allowing to change one without having to change the other. Large processes are broken down into smaller ones ending in networks of enterprise activities which are connected by the behavioural rule sets. It is this network of enterprise activities which represent the business process model to be used in the operational support (Kosanke, Vernadat, & Zelm, 1999).

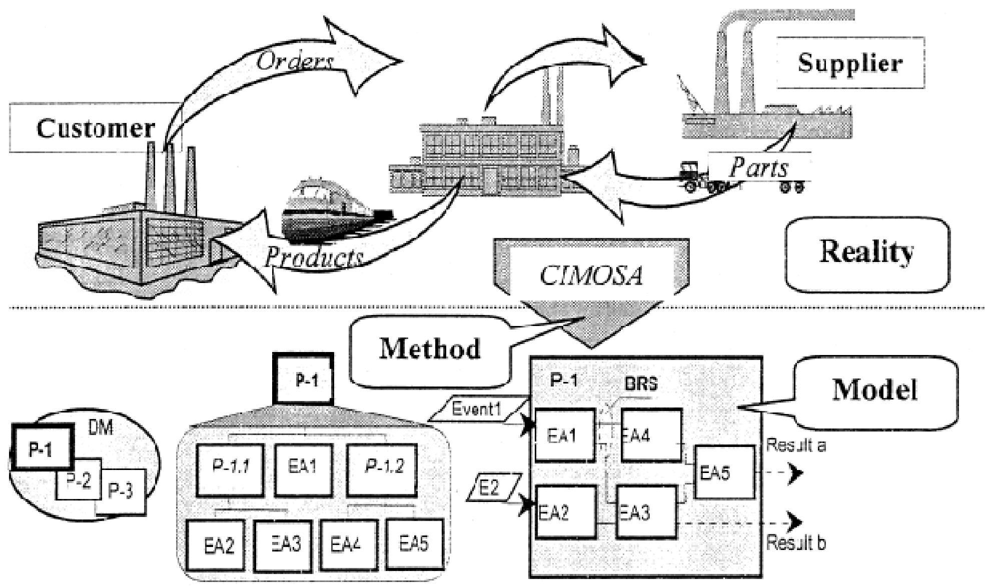


Figure 3.6 CIMOSA process functionality and behaviour.
Reference: (Kosanke, Vernadat, & Zelm, 1999).

The entire enterprise is modelled according to its object classes, events, business processes, domain processes, activities and elements. Figure 3.6 represents the modelling framework with components in abbreviations. These components have a hierarchical modelling structure. Figure 3.7 explains each component of view and their hierarchical structure.

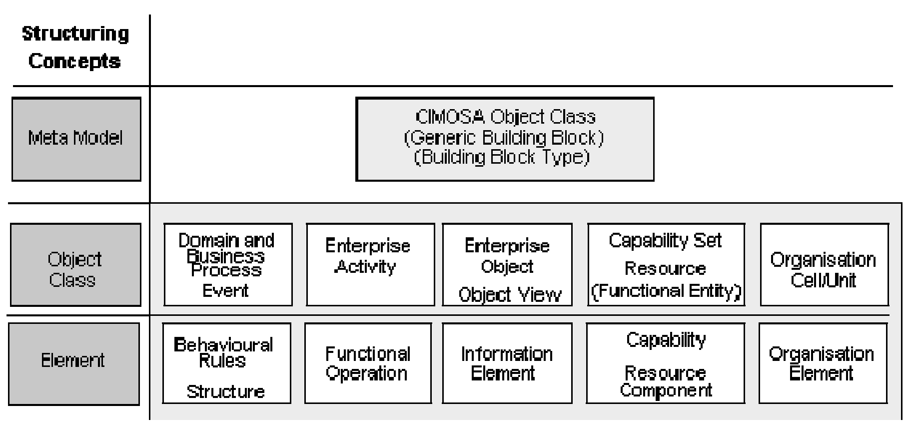


Figure 3.7 CIMOSA modelling components.
Reference: (CIMOSA, 1999)

CIMOSA does not provide a detailed methodology for CIM system design but recognizes the coexistence of several methodologies (to be developed according to business user needs by CIM users, or by CIM consultants). However, CIMOSA defines a generic CIM system life cycle as a sequence of phases to be used to build the particular architecture of a CIM environment, from requirements definition to system installation, test and release, and later on, system maintenance. Only well-defined pieces of the methodology have been documented in the CIMOSA Technical Baseline (CIMOSA, 1996) because such methodologies are not unique or universal.

The CIMOSA system life cycle comprises the following major phases:

- **master plan definition:** definition of overall business objectives, constraints, and guidelines for organizational structure - no construct provided;
- **requirements definition**: precise definition of all business processes and enterprise objects for each enterprise domain;
- **system design:** detailed specification and implementation description of all enterprise activities with time, resource, exception handling, and organizational requirements as well as information system structures for each business process.

CIMOSA provides four ways for process synchronization (Berio & Vernadat, 1999):

- synchronisation by events (one activity in a process P1 generates an event Ev1 which triggers another process P2, either in the same domain or in another domain.),
- synchronisation by object availability: the output of an activity of process P1 can be the input of an activity of process P2,
- synchronisation by resource availability (resources are allocated to processes on the basis of schedules or priority rules.),
- synchronisation by message passing.

In accordance with CIMOSA modelling framework as explained in Figure 3.6 and 3.7, Figure 3.8 represents a sample process modelling showing an order process.

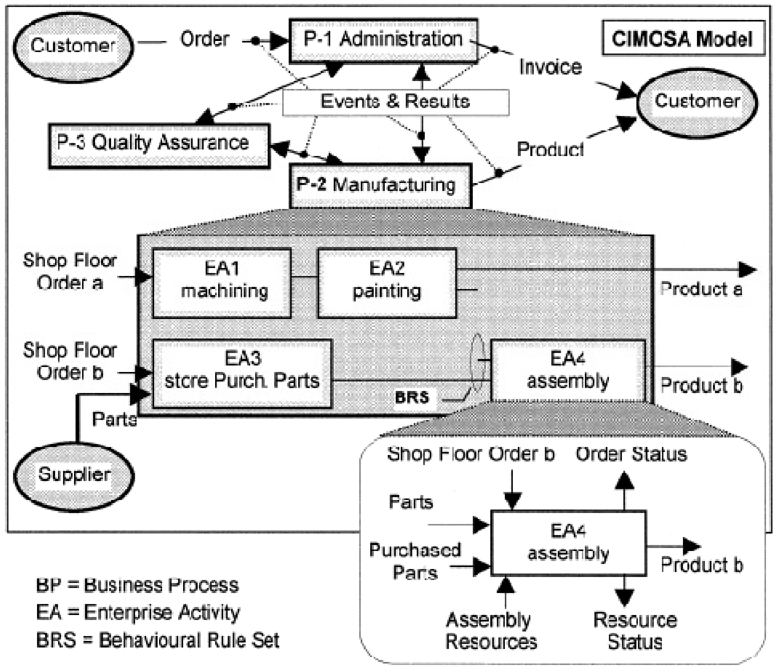


Figure 3.8 Order processing example.
Reference: (Kosanke, Vernadat, & Zelm, 1999)

CIMOSA has been applied in many modelling processes, e.g. operational representation of business processes (Carla, 1999), XML integrated modelling (Salvato, Leontaritis, Winstone, Zelm, Rivers-Moore, & Salvato, 1999); Prime Object Tool is integrated to the general structure of CIMOSA (Bruno & Torchiano, 1999); CIMOSA- compliant tool is developed called First Step which is proven to be one of the most applicable business process management tools (Levi & Klapsis, 1999); OPAL execution environment (Solte, 1999); and CIMOSA implementation of product design process based on the methodology "quality function deployment" (Chin, Lam, Chan, Poon, & Yang, 2005). Another specific process modelling is applied for production planning using CIMOSA constructs in (Ortiz, Lario, Ros, & Hawa, 1999). Process design of manufacturing cells are closely related to the computer integrated manufacturing and naturally to the CIMOSA applications (Monfared & Weston, 1999) .CIMOSA can be integrated with decision modelling tools, e.g. simulation or Petri nets. For instance, Wilson, Aguiar, Edwards (1999) is about implementation of CIMOSA for manufacturing business processes combined with Petri nets in order to obtain process integration.

In this section, CIMOSA constructs has been explained and presented in more detail than the others. There are two reasons for this; the first one is that CIMOSA is one of the most detailed enterprise reference architectures especially for the manufacturing processes including computerized process management systems. The second reason is that this thesis study for requirements analysis employs the CIMOSA constructs.

3.5 GIM (GRAI-IDEFO-Merise) and GRAI

GIM originally denotes to GRAI-IDEFO-Merise, as a methodology for analysis and conceptual design of manufacturing systems (Roboam, Zanettin, & Pun, 1989). Since then, the name has been changed to GRAI Integrated Methodology (Doumeingts, Vallespir, Zanettin, & Chen, 1992).

GIM has its origins in GRAI, *Graphes a Resultats et Activites Interrelies*, which is a method to model and analyze automated manufacturing systems, and in Merise, an information system design and analysis methodology widely used in Europe. Both GRAI and GIM have been developed at the University Of Bordeaux, France. The development of GIM has been partially funded by the ESPRIT program of the European Communities (EP 418 and EP 2338) (Vernadat, 1996, p.45).

At the roots of both GRAI and GIM is a conceptual model called the GRAI conceptual model (Figure 3.9), borrowed from the general system theory and systems organization theory. This model is also at the roots of Merise. The model says that any enterprise, like any complex dynamic system, is made of three fundamental sub-systems: a physical system, an information system, and a decision system. GRAI also adds an operating system.

- The **physical system** transforms the material flow. It is made of work stations or cells involving machines, workers, parts, etc.
- The **operating system** is dedicated to real-time control of the physical system.
- The **decision system** is the locus of decisions for the whole enterprise via a hierarchical structure organized into decision levels made of decision centers.
- The **information system** makes the link between the decision system, the physical system, and the enterprise environment. It transforms and memorizes information.

The GRAI method is based on a methodology for analyzing manufacturing systems. It makes use of two basic modelling tools which are denoted by GRAI grid, and GRAI nets.

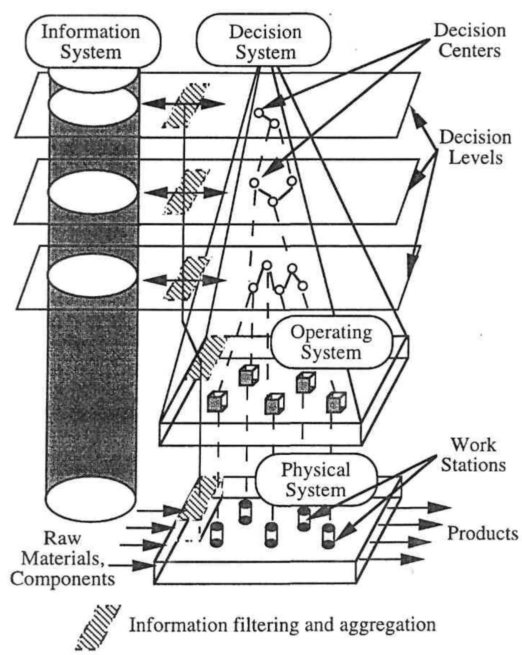


Figure 3.9 The GRAI conceptual model.
Reference: (Vernadat, 1996, p.49)

The **GRAI grid** is used to perform a top-down analysis of the domain of the enterprise to be analyzed. It is made of a two-dimensional matrix in which columns represent functions, and lines represent decision levels defined by a horizon *H* and a period P $(H = n.P,\ n > 1)$. Long-term planning horizons are at the top, and short-term levels are at the bottom of the grid. Functions are the usual functions of an enterprise (such as to design, to plan, to produce, or to sell). One or more columns are reserved for information. Each cell in the matrix defines a **decision center** (Figure 3.10). The grid is then used to analyze relationships among decision centers in terms of flows of information and flows of decisions

GRAI nets are used to further analyze decision centers in terms of their activities, resources called supports (information or mechanisms), and input/output objects. In this way, a bottom-up analysis of the manufacturing system studied can be made to validate the top-down analysis. In practice, several paths in both ways are necessary to converge to a final model accepted by all business users concerned (Vernadat, 1996).

Functions / H/P	Internal and External Information	To design	To plan	To produce	To sell
H1 = P1 =					
H2 = P2 =			Decision Center		
H3 = P3 =					

Figure 3.10 The GRAI grid.
Reference: (Vernadat, 1996, p.50)

GIM integrates many methods and tools in one framework as given in Figure 3.11. GIM and GRAI have common properties that their activity modelling is carried out using IDEF0 tools and information modelling is carried out using Merise tools and methods. Other operational analysis and data modelling tools can be integrated to GIM Framework.

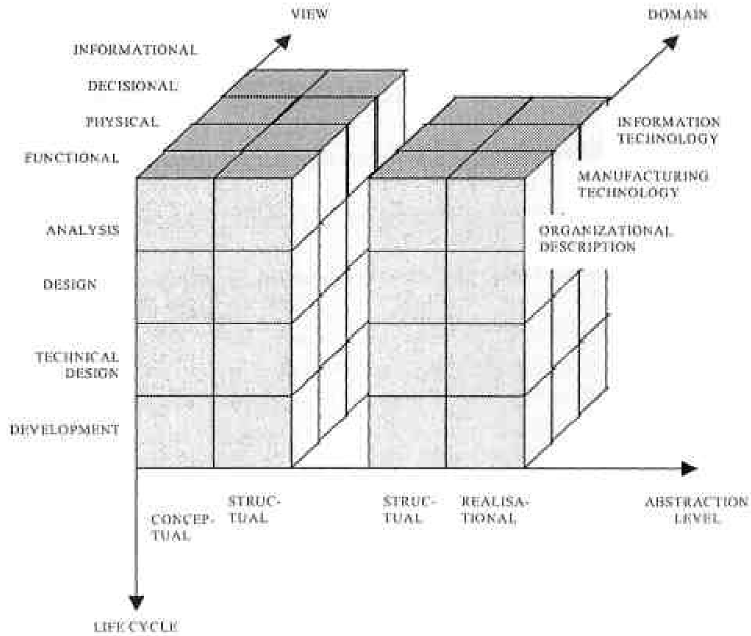


Figure 3.11 GIM modelling framework.
Reference: (Doumeingst, Ducq, & Kromm, 1999)

The structure of GIM modelling framework is illustrated in Figure 3.11. As shown in the GIM modelling framework, an enterprise can be described using four views; functional, physical, decisional and informational. The GIM has a cubic structure with modelling views, lifecycle dimensions, and abstraction levels. The life cycle of the GIM has the following phases: analysis, design, technical design, and development. The three abstraction levels are: the conceptual, structural, and realizational. The conceptual level answers the question 'what' without any organizational or technical consideration. The structural level answers the questions 'who', 'when', and 'why' to integrate an organizational point of view. The realization level asks the question 'how' to integrate technical constraints (Doumeingts, Vallespir, Zanettin, & Chen, 1992). Figure 3.12 includes the modelling levels with views.

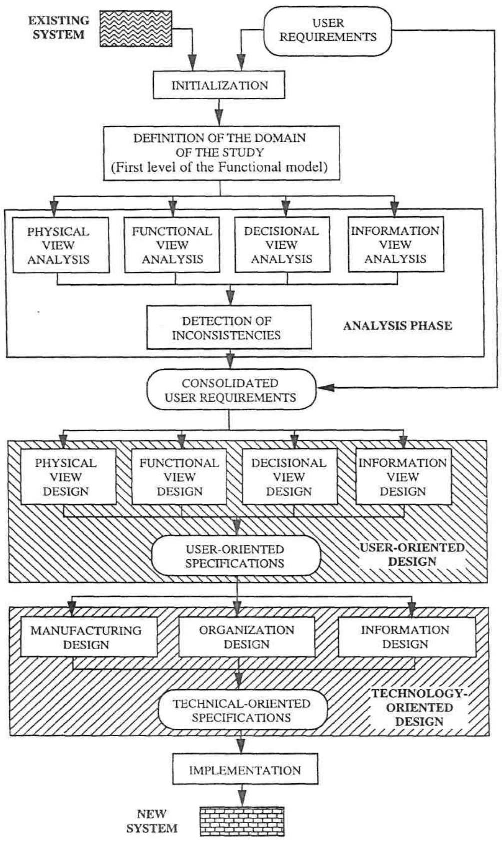


Figure 3.12 GIM structured approach.
Reference: (Vernadat, 1996, p.54)

GRAI and GIM are supported by a structured methodology. The goal is to provide specifications for building a new manufacturing system in terms of organization, information technology, and manufacturing technology viewpoints. The methodology includes four phases (Figure 3.12): initialization, analysis, design, and implementation. In each phase before implementation, similar views are described as in the other architectures.

The Functional view shows the main functions of the manufacturing systems and flows between the functions (Berio & Vernadat, 2001). There are three types of functional activities namely; product management activities, planning activities, and resource management activities. IDEFO methodology is used to represent the models in the functional view (Yoshikawa & Goossenaerts, 1993). This view similar to CIMOSA describes the functions (activities) and the behaviour (flow of control).

The physical view is composed of people, facilities, materials, equipment, and machines, which focus on transforming raw materials and components into final products in order to add value to the material flow. It can be categorized into process controlled and performance based. IDEF0 methodology is used to represent the models in this view (Yoshikawa & Goossenaerts, 1993). This view is similar to the resource view of CIMOSA. In addition to information about the resources this view also considers the material (resource) transformation (Venugopalan, 2003).

The decisional view is the locus of decisions for the whole enterprise via a hierarchical structure organized into decision levels made of one or more decision centers. The decisional view has two main parts: periodic-driven and event-driven. The periodic-driven part forms the basis for high-level decision making. The event-driven part interfaces with the physical view and consists of numerical control systems, programmable controllers, and other operating systems (Yoshikawa & Goossenaerts, 1993). This view covers the functional aspect of making decisions, hence it is classified under the functional view of CIMOSA.

The Information view contains all the information that a decisional view requires and is structured hierarchically. The information view makes the link between the decision view, physical view and the enterprise environment. Entity relationship method is used to represent models in this view (Yoshikawa & Goossenaerts, 1993). The information views of CIMOSA and GIM are similar as they depict the information requirements in an enterprise.

3.6 IDEF Modelling

Enterprise modelling methods, architectures and tools can be used in support of the life cycle engineering of large scale, complex and changing systems (Kosanke & Vernadat, 1998). The IDEF suite of enterprise modelling approaches, which comprises IDEF0, IDEF1, IDEF1x, IDEF3 and other graphically based modelling notations (Ang, Peng, & Keng Leng,1999) have been applied extensively in support of large industrial engineering projects. Individually, these notations are designed to model an enterprise from a defined viewpoint, such as a ‘‘function viewpoint’’ or an ‘‘information viewpoint’’. This is both strength and a weakness of IDEF enterprise modelling approaches. However, possibly because IDEF modelling concepts and tools have been incrementally developed over a number of decades, there is no overarching modelling framework that has been formally defined to interconnect individual IDEF notations. Each can be individually applied and reapplied, in a variety of ways and its use can be supported by a selection of proprietary systems engineering tools.

The building block of this methodology is the Activity box as shown in Figure 3.13 and Figure 3.14. The box defines a specific manufacturing activity in the manufacturing process. The Activity may be a decision making activity, an information conversion activity or a material conversion activity. The Inputs are the items that are transformed by the Activity, and the Output is the result of the Activity. A Control is a condition needed to perform the Activity. The Mechanism is the means by which the Activity is realized. The boxes together with their interfaces (Input, Output, Control and Mechanism) form the Diagrams of the methodology. In addition, the methodology also includes Texts and Figures which are used to supplement the diagrams. The former uses texture descriptions to elaborate a diagram, while the latter use figures.

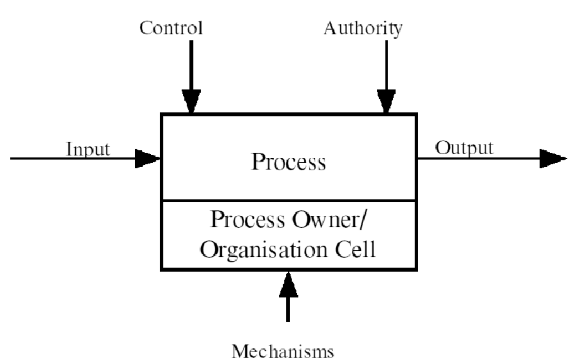


Figure 3.13 IDEF* process notation.
Reference: (Ang, Peng, & Keng Leng, 1999)

IDEF0 uses a top-down decomposition to break up a complex topic into small pieces which can be more readily understood and which are set in their proper context with respect to system elements. It provides the ability to show what activities are being carried out within a process, what connects the activities and what constrains the activities. It uses a structured set of guidelines based around hierarchical decomposition, with excellent guidance on abstraction at higher levels. Using IDEF0 as a modelling technique ensures that the content for any part of a process model under analysis in relation to the whole of the process model is always known (Vernadat, 1996, p.128)

In manufacturing systems design, it is generally accepted as a good practice to construct an 'as-is' model of a manufacturing system as the first step to understand and change the system. An IDEF0 'as-is' model provides a means of examining the relationships between activities in order to evaluate how a modification in an activity may impact on other activities to influence the performance of the overall system. It therefore forms the basis for the development of the 'to-be' model which defines a strategy for change or goal (Ang, 1999).

IDEF0 was developed in order to represent activities or processes (comprising partially ordered sets of activities) that typically are carried out in an organised and standard manner (FIPS-183, 1993). The IDEF0 definition of a function is ''a set of activities that takes certain inputs and, by means of some mechanism, and subject to certain controls, transforms the inputs into outputs''. These inputs, controls, outputs and mechanisms (ICOMs) can be used to model relationships between different activities.

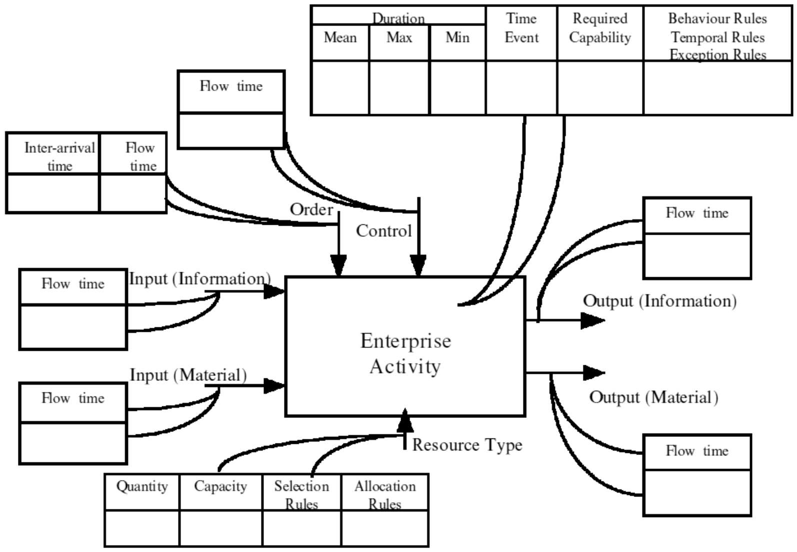


Figure 3.14 Pop-up boxes of information inputs in IDEF.
Reference: (Ang, Peng, & Keng Leng, 1999)

The IDEF0 model of a business describes the functions performed by the business process and their interfaces, namely inputs, outputs, controls and mechanisms. Although these interfaces can be information or physical objects, they are represented only at the level of graphical labels and no actual information structures are attached to those labels. An IDEF1x model of the business process on the other hand graphically represents the information content and structure related to a business process or an enterprise system. It is possible to derive an IDEF1x model from an IDEF0 model by using the Glossary of the IDEF0 model as the entity pool for IDEF1x. A prerequisite is that the IDEF0 model must be of sufficient detail to enable all of the possible candidate attributes and entities to be identified (Ang, Peng, & Keng Leng, 1999).

The IDEF3 notation was developed as a means of describing the time-based behaviour of systems (Mayer, Menzel, Painter, DeWith, Blinn, & Perakath, 1995) and provides means of representing sequence, timing and reachable states as presented in Figure 3.15. IDEF3 provides two main groups of modelling mechanism, namely: Process Flow Network modelling constructs and Object State Transition Network modelling constructs. Process Flow Networks represent the order in which, and conditions under which, activities are performed by a system. The Object State

Transition Network describes the ''transition states'' that an object can pass through during the execution of a specific process (Kim, Weston, Hodgson, & Lee, 2003).

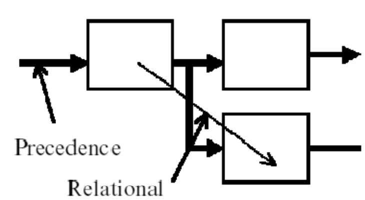
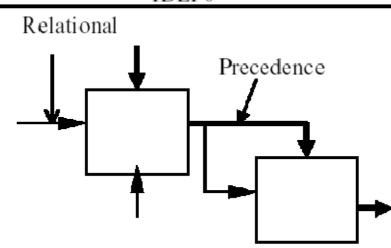
	IDEF3 (PFD)	IDEF0*
Links		
Join/Split logic	***AND, OR, XOR***	***AND, OR, XOR***
Hierarchy	Hierarchically decomposed diagrams.	Hierarchically decomposed diagrams.
Text	Text for each diagram.	Text for each diagram
Glossary	Glossary of terms used in diagrams.	Glossary of terms used in diagrams.

Figure 3.15 Comparisons of IDEF3 and IDEF0*.
Reference: (Ang, 1999)

Before the1990s the concept of enterprise modelling and engineering are not spreadly known and applied in literature and practical industrial studies. Those days' industrial applications are based on reengineering and process modelling studies. Later on those studies have prepared the infrastructure of enterprise modelling. Process modelling and reengineering task need structural and formal modelling constructs, and IDEF0 (Ang, 1999) is one of these constructs which is popularly applied (Zakarian & Kosiak, 2001). Enterprise modelling concept then arises and IDEF3 (Plaia & Carria, 1995) has been developed by improving IDEF0 so that enterprises are modelled by integrating the other perspectives within the enterprise, i.e., informational, resource, and organizational perspectives. Shop floor activities are modelled by IDEF0 constructs; data flows are modelled by using IDEF1x constructs; and the dynamic message flows and specifications based on the message requirements on the function mode l are built using a variant IDEF3 process modelling method (Cho & Lee, 1999). GRAI –IDEF integration is the subject of many enterprise modelling studies and generally simulation schemes are developed to analyze the integration (Al-Ahmari & Ridgway, 1999). During process or enterprise requirements analysis engineers try to apply requirements modelling tools, e.g. UML (see chapter 4) which is commonly used in software engineering applications. These requirements analysis tools are generally used to model the data and information flows integrated with the processes. IDEF as an enterprise engineering tool can also be integrated with UML for detailed analysis (Dorador & Young, 2000) so that the enterprise model can be the meta-model of further enterprise applications. IDEF process modelling provides a successful infrastructure in terms of the applications in quality standards, especially for ISO 9001 (Gingele, Childe, & Miles, 2002).

Enterprise models play a crucial role in the analysis of enterprises and constitute a basis for improvement or reengineering. A great variety of description techniques exists, among them IDEF0 for function modelling and IDEF1x for information modelling. Verification of these models is important but there are not so many studies on this subject. IDEF1x uses the data coming from IDEF0, so verification is needed for each phase of modelling, e.g. one in IDEF0 and one in IDEF1.. (Kacprzak & Kaczmarczyk, 2006) proposes such a model for only one phase of the modelling.

IDEF modelling notations were designed to provide means of modelling enterprises in their entirety, so as to systematically deliver abstract representations of different enterprise views that can be used by concerned parties in different ways.

3.7 PERA

The Purdue Enterprise Reference Architecture (PERA) and the related methodology have been developed at the University of Purdue since 1989 on the basis of previous work in the area of CIM by a team led by Prof. Williams at the Purdue Laboratory for Applied Industrial Control (Williams, 1994).

PERA is covered by very simple graphical formalisms and easy-to-understand textual manuals because it has been designed for non-computer science educated users. Indeed, users must be able to apply the methodology themselves to their enterprise (or the part or system to be analyzed). The methodology starts first with the identification of the enterprise entity, i.e., the part of the enterprise to be considered. This is done by the corporate management. Then, the enterprise mission is defined in terms of products or services to be offered. The third phase, or definition phase, defines basic requirements for manufacturing personnel and information policies on one side, and product and manufacturing units on the other side. The specification layer defines functional requirements, i.e., instrumentation and control diagrams, management and union-mandated requirements, and plant layout. The detailed design layer is concerned with detailed physical design, i.e., equipment selection, definition of personnel skills, organizational planning, training programs, and plant facilities layout. The manifestation layer corresponds to the plant installation, i.e., equipment installation, staffing, training, plant construction, testing, and commissioning. The plant is then ready for operation. The operation layer corresponds to the day-to-day exploitation of the plant and continuing process development and maintenance. It will end with plant obsolescence (Vernadat, 1996, p. 55).

PERA is defined by its structure with layers (Figure 3.16) which has been developed to cover the full enterprise life cycle from inception and mission definition down to its operational level and final plant obsolescence. Each layer defines a task phase. Each phase is informally described by a technical document as a set of procedures for leading a user's application group through all the phases of an enterprise integration program.

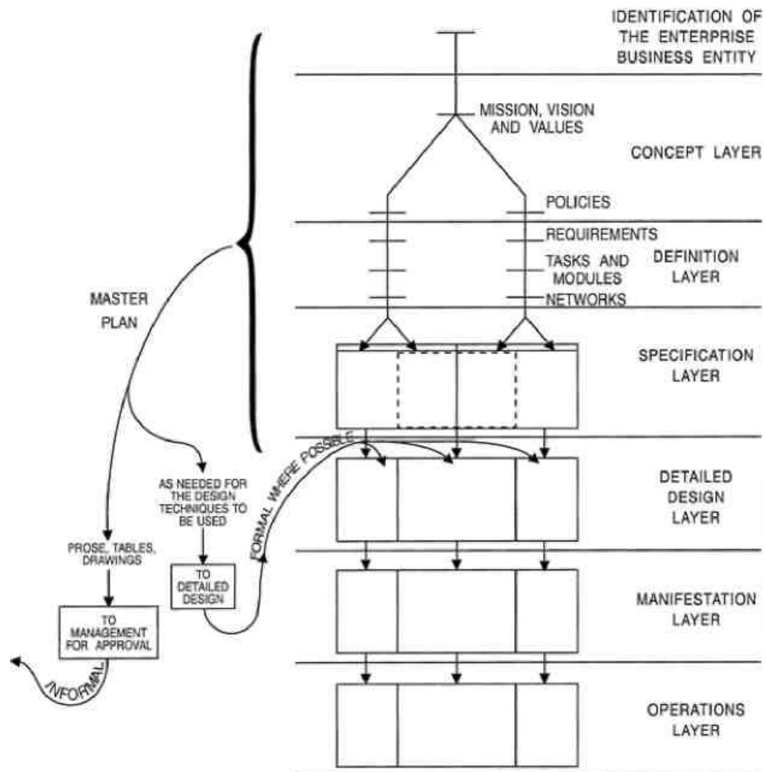


Figure 3.16 Modelling structure of PERA.
Reference: (Vernadat, 1996, p.56)

As pointed out by PERA, every enterprise must have a mission in order to justify its existence. In this regard, several separate enterprises may share a set of common goals among themselves while each has its own specific goals as well. For example, individuals and departments within a single company may have to address some particular requirement or task of their own in addition to the coordination required between them (Williams, 1994). Figure 3.16 shows how PERA separates a single overall Enterprise Entity into several sub-entities by going through a separation of the overall mission into its different components. Each of the separate organizational entities of the enterprise or sub-unit of another larger enterprise would have its own architecture as described by PERA (Li & Williams, 2000).

PERA focuses on two main views, namely the functional and implementation view as opposed to CIMOSA, which focuses on four views, namely, function, information, resource and organization views. Each view of PERA is structured along: (1) an information stream, which is initiated by planning, scheduling, control and data requirements of the enterprise, and (2) manufacturing related stream, which is initiated by the physical production requirements of the enterprise (PERA, 1999). Both of these views are discussed in the following paragraphs. The "extent of automation" line in Figure 3.17 defines the actual degree of automation carried out. It shows the split in assignment of functions between human and physical equipment. The human and organizational architecture interfaces between humans. The manufacturing architecture interfaces between various manufacturing equipments excluding computers and defines all the tasks performed by plant equipment. The information architecture defines filling tasks performed by the computers, software, and databases. It interfaces between information equipment including computers (Williams, 1998).

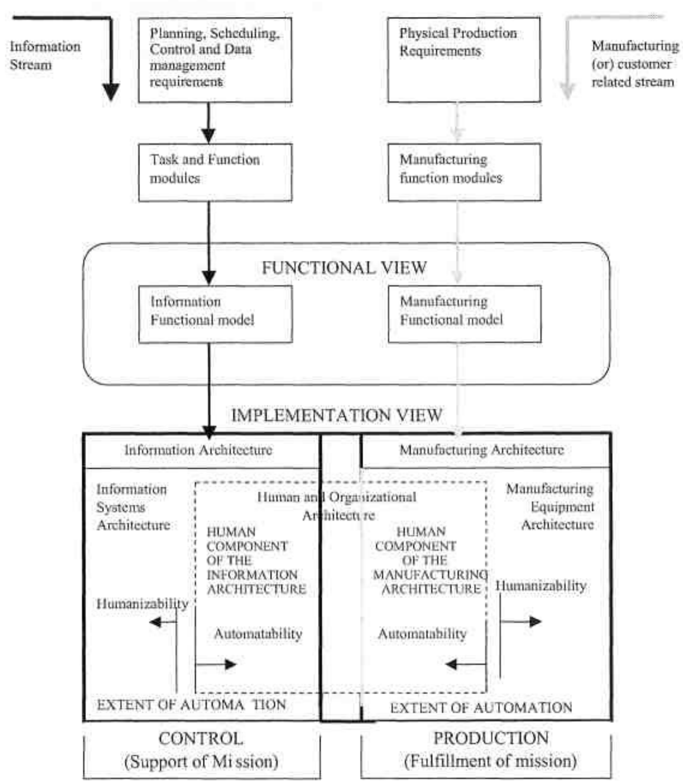


Figure 3.17 Views and architecture of PERA.
Reference: (Tham, 2001)

The functional view refers to a collection of task modules (including the interconnectivity) that describes and illustrates the functions assigned to a business entity. The functional view refers to a collection of task modules (including the interconnectivity) that describes and illustrates the functions assigned to a business entity and their relationship to each other. This view is composed of the informational functional model and the manufacturing functional model, which respectively belong to the information stream and manufacturing related streams. The information functional model provides the input to define the information architecture of the implementation view; the manufacturing functional model serves as the input for the manufacturing architecture in the implementation view (Nagarajan, Whitman, & Cheraghi, 1999). This view is similar to the function view in CIMOSA in that it describes functions but it also takes the informational part into account in the same view. This view does not mention the behaviour or control aspects as in CIMOSA (Venugopalan, 2003).

The Physical or Implementation View is a collection of the human organizations and the physical hardware and software that is used to carry out all or part of the functions that are described and illustrated in the functional view of a business entity. It is composed of the information architecture, human and organization architecture and manufacturing architecture (Nagarajan, Whitman, & Cheraghi, 1999). This view represents the humans, hardware and software capabilities and is similar to the resource view in CIMOSA. It also contains information on application programs and software. This view gives more focus on the human component and identifies the extent to which humans can perform the tasks.

Purdue methodology is good at engineering designs. On the other hand, the lack of sufficient theoretical studies also remind us that the success possible from an application of a systems engineering technique will largely depend upon the human understanding, judgment and decision making involved. Methodologies also will surely help but never more than to the extent that an engineering drawing helps its users (Li & Williams, 2002). PERA does not provide its own modelling tools. It can be used in connection with any other existing technique for modelling enterprise aspects. PERA is ready to be integrated with other architectures as a complementary framework. PERA is extendable to be applied in various industrial sectors. Even if it was originally developed with manufacturing systems, it can be used for all types of industries (Vernadat, 1996, p.57).

3.8 ARIS

ARIS means Architecture for Integrated Information Systems. It has been developed by Prof. Scheer at the University of Saarbrücken in Germany (Scheer & Kruse, 1994). Its overall structure is very similar to CIMOSA, but instead of focusing on computer-integrated manufacturing systems, it deals with more traditional business-oriented issues of enterprises such as order processing, production planning and control, inventory control, etc. The focus is essentially on software engineering and organizational aspects of integrated enterprise system design. Figure 3.18 provides a global view of the architecture. It is structured into four views and three modelling levels (Vernadat, 1996, p.58).

This architecture is mainly used to describe enterprises and application software. The derivation of the architecture is carried out with a business process perspective of the enterprise. Process chains are important support for business information systems.

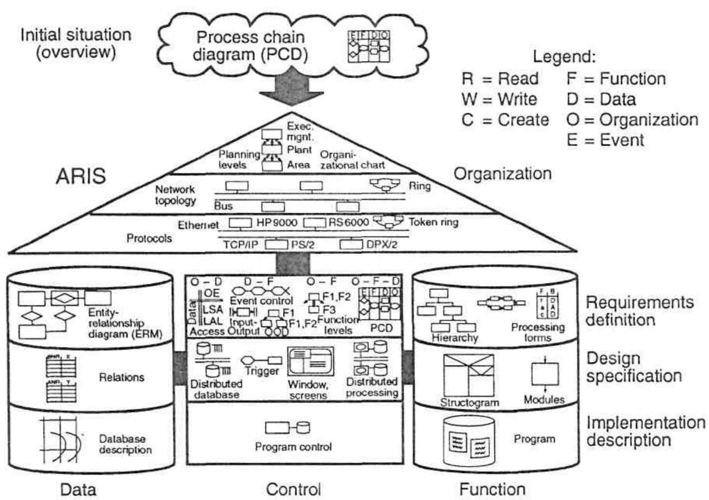


Figure 3.18 Structure of ARIS.
Reference: (Scheer & Kruse, 1994)

Figure 3.19 illustrates ARIS architecture with its four modelling views (data, control, function and organization) and its three modelling levels (requirements definition, design specification and implementation description). ARIS has the same modelling levels as CIMOSA's derivation principle. In the ARIS architecture another view, namely the control view, has been introduced in order to maintain the relationship between the other views - data, function and organization. It should be noted that the resource view is missing in this architecture. The resource view of information system is very broad and includes components such as the CPU, peripherals, networks, and programming and database systems. The resource view is considered at the descriptive levels of design specification and implementation of other views - function, organization, data and control. Therefore the resource aspect is considered under other views and is not considered as an independent view (Yoshikawa & Goossenaerts, 1993).

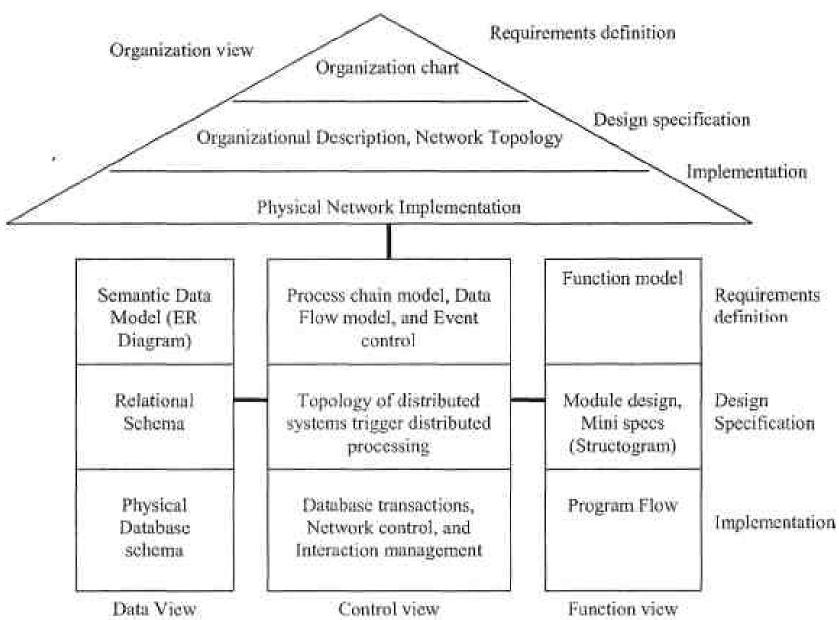


Figure 3.19 ARIS Architecture with its views.
Reference: (Venugopalan, 2003)

The function view provides a description of process rules and process structure. In ARIS the terms “process” and “function” are used interchangeably. As shown in Figure 3.16, this view is used to define the function model at the requirements definition level. The function model is translated to a module design and structogram, and program codes at the design phase (Vernadat, 1996, p.59). This view attempts to answer this question, which functions will be performed by the enterprise? Some examples include production plan creation, and order processing. The function view takes the control aspect into account at design specification level using a structogram, similar to CIMOSA.

The Data View represents events, status of events, and environmental conditions. They are represented as information objects using data view (Yoshikawa & Goossenaerts, 1993). As shown in Figure 3.19, this view is used to define semantic data models (ER diagrams) -at requirement definition level, which is translated into relational schema at the design specification level, and finally implemented using the physical database at the implementation level (Vernadat, 1996, p.58). This view answers the question of what information is important to the enterprise. Some examples are customer, supplier, product, and material information.

Both the users and the organizational unit are aggregated into a single element and represented in **the organization view.** That is, users are assigned to respective organizational units on the basis of such criteria as same function or same work object (Yoshikawa & Goossenaerts, 1993). This view is used to define the enterprise structure in terms of organization chart at the requirement definition level that is translated into network topology and physical network implementation at the design specification and implementation levels, respectively. This view also covers the resource aspects (Vernadat, 1996). This view answers which organizational units exist in the enterprise. Some examples include sales, purchasing, and accounting. This view, similar to CIMOSA, consists of information about responsibilities within the enterprise. In addition, it determines who is responsible for doing what by means of assigning users with organizational units.

The main purpose of **the control view** is to maintain the links and relationships between the other three views (data, function and resource). The integration of these relationships in a separate view makes it possible to systematically enter all the relationships without any redundancy. The business processes are put together and implemented as a logical chain in this view (Yoshikawa & Goossenaerts, 1993). The control view is where the business processes, also called activity chains, can be put together and implemented as logical sequences of program execution with relevant computer screens and distribution of data over the enterprise network (Vernadat, 1996, p.59). ARIS has a separate control view, unlike CIMOSA, where it is within the function view.

ARIS is an open architecture in the sense that the formalisms used within the various views and levels of the architecture are not fixed forever. The architecture is populated by the best methods currently available, in the ARIS designers' opinion. This set may be updated or expanded when new methods have proved their value. The only criterion considered is compatibility and reduced overlapping of these techniques to form a consistent structured engineering approach.

The ARIS architecture is now supported by a tool also called the ARIS-Toolset, and is being used in industry, mostly in Germany. ARIS is being applied for business process re-engineering of managerial information systems (Scheer & Kruse, 1994).

3.9 GERAM

GERAM stands for Generic Enterprise Reference Architecture and Methodology (Bernus & Nemes, 1994). It has been developed by the aforementioned IFAC/IFIP Task Force as a generalization of the architectures presented in the previous sections. The Task Force was formed at the IFAC World Congress in Tallinn, Estonia, in August 1990 with the mission to study the field of enterprise reference architectures for the purpose of picking the best one, or, if no one can be found, propose a method for the development of a better one. GERAM essentially builds on results from CIMOSA, GIM, and PERA. Upon completion, it will be submitted to international standardization bodies such as ISO for consideration.

Previous research carried out by the AMICE Consortium on CIMOSA, by the Purdue Consortium on PERA, and by the GRAI Laboratory on GIM, (and similar methodologies by others) has produced reference architectures which were meant to be organising all enterprise integration knowledge and serve as a guide in enterprise integration programs. GERAM is about those methods, models and tools which are needed to build the integrated enterprise. The architecture is generic because it has the potential for application to most types of a real or virtual enterprise. GERAM is expected to comprise seven major components as follows (Bernus & Nemes, 1997):

- Generic Enterprise Reference Architecture (GERA);
- Generic Enterprise Engineering Methodology (GEEM);
- Generic Enterprise Modelling Languages (GEMLs);
- Generic Enterprise Modelling Tools (GEMTs);
- Generic Enterprise Models (GEMs);
- Generic Enterprise Modules (GMs); and
- Generic Enterprise Theories (GTs).

The Generic Enterprise Reference Architecture (GERA) will provide the definition of enterprise related concepts with a primary focus on the enterprise life cycle. The life cycle of an enterprise is modelled by a matrix representation as given by Figure 3.16, on one axis are the development steps of the enterprise integration program (identification, concepts, requirements, design, implementation, build, and operations). This structure follows the procedure proposed by PERA and in each phase pays attention to machine and human aspects. On the other axis, is the genericity axis of CIMOSA or the ENV 40 003 (generic, partial, particular levels) with the views (function, information, decision/organization, resource/structure). With this matrix, it is possible to compare and evaluate the architectures presented in the previous sections. For instance, CIMOSA completely fills the requirements, design, implementation, build, and operations level but not the identification and concept levels, while PERA completely fills the particular level from top to bottom. This shows the complementary nature of CIMOSA and PERA and identifies a gap for basic constructs for generic and partial levels at the concept level.

Generic Enterprise Engineering Methodology (GEEM) is the description, on a generic level, of the processes involved in an enterprise integration program. This will be a set of well-documented, detailed process models with user guidelines provided for each step. At the current state of GERAM, the best candidates for GEEM are PERA and the GIM methodology.

Generic Enterprise Modelling Tools and Languages (GEMTs and GEMLs) will be a set of recommended languages and tools which can be used for enterprise engineering. Several tools or languages can be recommended for the same purpose. Indeed, the engineering of an integrated enterprise is a complex, multi-disciplinary management, design, and implementation exercise during which various forms of models of the target enterprise need to be created. The final choice of the tools is left to the user. In the current version of GERAM, proposed GEMTs and GEMLs are the IDEF suite of modelling methods complemented by IDEF3, the CIMOSA modelling language, and the **GRAI** grid.

Generic Enterprise Models (GEMs) capture concepts which are common to all enterprises. Therefore, the enterprise engineering process can use them as tested components or partial models for building any specific enterprise model.

Generic Enterprise Modules (GMs) are **products,** which are standard implementations of components that are likely to be used in enterprise integration, either by the enterprise integration project, or by the enterprise itself. Generic modules can be configured to form more complex modules for the use of an individual enterprise.

Generic Enterprise Theories (GTs) describe the most generic aspects of enterprise-related concepts. Generally called ontological theories or simply ontologies, they may also be considered as 'meta-models' because they consider facts and rules about the facts and rules of the enterprise models.

GERAM modelling framework is shown in Figure 3.20. As shown in the figure, it consists of three dimensions: instantiation, life cycle phases, and views:

- The instantiation dimension consists of the generic, partial and particular architectures. This dimension is similar to the instantiation principle of CIMOSA, where a generic layer consists of basic building blocks with constraints, rules and functions. A partial layer consists of partial models. A particular layer is a model of a specific enterprise built from the generic and partial layers.
- The life cycle phases, including identification, concept, requirements, design (includes preliminary and detailed design), implementation, operation, and decommission.
- The view dimensions that are categorized mainly according to activity, physical manifestation, model content and implementation.

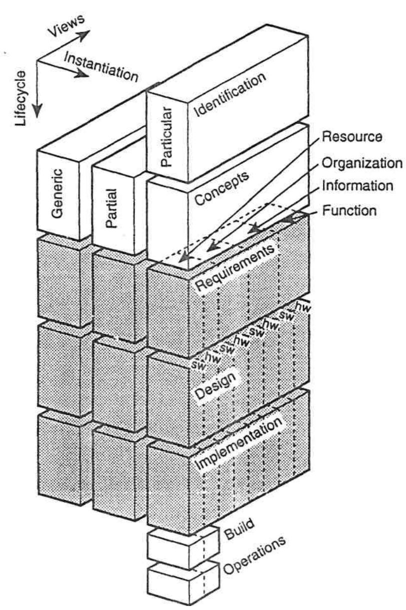


Figure 3.20 GERAM representation.
Reference: (Bernus & Nemes, 1994)

As shown in Figure 3.20 GERAM identifies three modelling dimensions of which the life cycle and instantiation dimensions are very similar to the derivation and instantiation dimensions of the CIMOSA cube. However, the view dimension is quite different because in this, four model views coexist (model content, purpose, implementation, and physical manifestation). GERAM consists of six views in total which are explained based on (Bernus & Nemes, 1994) in the following paragraphs.

The function / decision view represents the activity (functionality) and the flow of control (behaviour) of business processes. This view includes the functional model, process model and the decisional models. Similar to CIMOSA, this view also depicts the functionality and behaviour of the processes. This view does not mention about the control aspect.

In the information view, the knowledge about the objects (material and information) contained within the enterprise are collected. The information is identified from the relevant activities and structured into enterprise information model. This view, similar to the CIMOSA's information view, contains the knowledge and information about the objects in the enterprise.

The resource view represents the humans as well as the technological components in the enterprise. Resources are assigned to activities according to their capabilities and are structured into resource models, e.g. for asset management.

The organization view identifies the responsibilities and authorities of all the entities that have been identified in the other views. Though the names of the modelling views in GERAM are the same as in CIMOSA, they are not identical according to (Bernus, 2001). GERAM is a generalization of the view concepts of much architecture, including CIMOSA, GRAI and others.

The entity purpose view represents the model content according to the purpose of enterprise entity and is composed of two different views: customer service and product, and management and control. Customer service and product view represents the contents relevant to enterprise entity's operation. The management and control view represents contents relevant to the management and control functions necessary to control that part of the enterprise entity.

The implementation view is divided into human activities and automated activities (similar to PERA). The human activities view represents all the information related to the tasks to be done by humans. The automated activities view presents all the tasks to be done by machines.

The physical manifestation of an enterprise entity is represented by two views: software, and hardware. The software view represents all information resources capable of controlling the execution of operational tasks in the enterprise. The hardware view represents all physical resources that have the capability to perform some sets of tasks in the enterprise, e.g. a computer system with given performance characteristics or an employee with given skills .

One advantage of GERAM is its enabling nature. Through the definitions of GERAM it is possible to meaningfully relate (and possibly combine) areas such as 'Business Process Re-engineering', 'Total Quality Management', 'Concurrent Engineering' etc.

At the same time, each of the reference architectures (life-cycle models for enterprises) continues to be developed (and undoubtedly others will join). The role of GERAM will then be to allow comparison of the advantages of particular reference architectures and the selection of the one that best matches a particular organization's requirements. It is equally possible that a particular organization selects parts of existing architectures and develops best match architecture for in-house use. This would of course not be possible unless a commonly acceptable framework of concepts - which GERAM is meant to be - is available.

GERAM shows how reference architectures can be applied to various subclasses of enterprise, e.g. product oriented- *vs.* project enterprise (e.g. manufacturing firm *vs.* a large engineering project as an enterprise), or real- *vs.* virtual enterprise (e.g. company *vs.* consortium). As an example, the life-cycle of an enterprise engineering project is described in GERAM in the same way as the life-cycle of the enterprise.

3.10 Zachman Framework

The Zachman Framework is a widely used approach for developing and/or documenting enterprise-wide information. This framework is based on practices in traditional architecture (art and science of designing buildings) and engineering. The purpose of the framework is to provide a basic structure, which supports the organization, access, integration, interpretation, development, and management of the organization's information systems. The framework can contain global plans, technical details, lists and charts, and natural language statements. Any appropriate approach, standard, role, method, technique, or tool may be placed in it.

This framework is represented as a matrix. The perspectives (planner, owner, designer, builder, programmer, and user) are represented as rows. The perspective represents the models that are of interest to a specific group of people in the enterprise, the abstractions/ views - What (Data), How (Function), Where (Network), who (People), When (Time) and Why (Motivation) are represented as columns. Each cell in the framework has a model associated with it, as shown in Table 3.2. It is evident that the top rows of the framework represent a higher level of abstraction than the lower rows and are things that are of much interest to the senior management. By contrast, the bottom row represents actual things, such as networks, people, computer programs and databases (Zachman, 1987).

Data is used to describe the composition of a system and is represented by means of a data model, namely an entity relationship model. Since data is a part of information, this view is classified under the information view of CIMOSA. *Function* represents a process and its focus is on how the process transformation occurs from one state to another. This view is depicted by means of a process model. This view translates the mission of an enterprise into more detailed definitions of its operations.

Table 3.2 Zachman framework

	DATA What	**FUNCTION How**	**NETWORK Where**	**PEOPLE Who**	**TIME When**	**MOTIVATION Why**	
Objective/Scope Contextual Role: Planner	List of things important in the Business	List of core business processes	List of Business locations	List of important organizations	List of events	List of Business Goals/ Strategies	**Objective/Scope Contextual** Role: Planner
Enterprise Model Conceptual Role: Owner	Conceptual Data/Object Model	Business process model	Business Logistics system	Workflow model	Master schedule	Business plan	**Enterprise Model Conceptual** Role: Owner
System model Logical Role: Designer	Logical data model	System Architecture model	Distributed systems architecture	Human interface architecture	Processing structure	Business role model	**System model Logical** Role: Designer
Technology model Physical Role: Builder	Physical data/class model	Technology design model	Technology architecture	Presentation architecture	Control structure	Rule Design	**Technology model Physical** Role: Builder
Detailed Representations (Out of context) Role: Programmer	Data definitions	Program	Network Architecture	Security Architecture	Timing Definition	Rule Specification	**Detailed Representations (Out of context)** Role: Programmer
Functioning Enterprise Role: User	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy	**Functioning Enterprise** Role: User

Reference: (Zachman, 1987)

Similar to the functional view, mentioned in other methodologies, the Zachman framework also defines the processes and the transformation of the products. In *network* abstraction, the main focus is on the flow of information and material between various components and where the flows/connections exist in the system. These flows are depicted by a network model. It is mainly concerned with the geographical distribution of the enterprise activities. Since this view focuses on the flow of information between the components of the enterprise, it is classified under the information view.

3.11 Other Architectures and Frameworks of Enterprise Modelling

Enterprise modelling projects can differ due to the nature of the enterprise environment. This situation enforces new framework developments. The other frameworks that are used in applications are presented as follows:

1) ARDIN: The IRIS Group, of the University Jaume I of Castellon, Spain has been working in the ARDIN research project since 1994. The objective is to develop and validate a step forward in the state of the art of the RA for Enterprise Integration to organize knowledge and experience obtained in our own architecture called ARDIN. This architecture is being built giving priority to its practical utility as project execution support in enterprise integration. A long range objective will be the achievement of the needed requirements and components to satisfy the GERAM requirements for a 'complete enterprise integration RA'. For the graphical representation of every process, different modelling tools are used inside ARDIN including IDEF0, GRAI Nets, and UML tools (Chalmeta, Campos, & Grangel, 2001).

2) Curtis Enterprise Modelling Framework: Curtis defines the views that are required for software process modelling. Some of the important forms of information included in a software process are: what is going to be done, who is going to do it, when and where will it be done, how and why will it be done and who is dependent on it being done. Curtis has identified four views for analyzing and presenting process

information: functional, behavioural, organizational and informational characteristics (Curtis, Kellner, & Over, 1992).

3) C4ISR (Command, Control, Communications, Computers Intelligence, Surveillance and Reconnaissance) Architecture Framework: C4ISR architecture includes a number of "products" that are in graphical, textual, and tabular forms. These products are considered either essential or supporting. Essential products are those that are required in order to develop the architecture framework; supporting products are those that provide data when needed depending on the objective and purpose of the specific architecture development effort. C4ISR provides a supporting framework. This framework consists of three views indicating the architecture views: operational view, system view, technical view) and a product number (C4ISR, 2002).

4) Presley Framework: This framework describes a modelling method, which supports the process-centered approach to meet the modelling needs of enterprise engineering. This method utilizes and integrates current modelling approaches and concepts-such as business process reengineering, enterprise engineering, IDEF suite of tools, object oriented concepts, semantic network and schema based representational schemes, and holonic and agent based representation methods- to build an integrated multi-view model of an enterprise. The views of an enterprise as discussed by (Presley, 1997) include: business rule/ information, activity, resource, business process, and organization.

5) Treasury Enterprise Architecture Framework (TEAF): The TEAF is structured as a matrix with views: functional, information, organizational, and infrastructure views. The TEAF uses these views from different perspectives covering planner, owner, designer, and builder. Work products document a set of related information for the TEAF. The work products take the form of documents, presentations, diagrams, charts, tables, matrices, or models (TEAF, 2000).

6) Shinkawa and Matsumoto Framework: Shinkawa and Matsumoto have identified five viewpoints; resource, organization, task, function and behaviour to

provide sufficient information about a business process. The first three represent the static aspect of the enterprise and the last two represents the dynamic aspect of the enterprise. The individual model units are then integrated into enterprise wide units using Rough Set Theory. The five viewpoints are defined grouped into static and dynamic aspects (Shinkawa & Matsumoto, 2001).

CHAPTER FOUR
REQUIREMENTS ANALYSIS AND MODELLING

4.1 Definition

Requirements are conditions or required capabilities that allow a user to solve a problem or meet an objective. From the system thinking perspective, a requirement is a condition or capability of a system or any of its components in order to satisfy a contract, normative, specification, or any other restriction formally stated in a document. There exist different definitions indicating the same concept of "requirements" organized in (Lopes & Barreto, 2002). The IEE Std 1233- IEEE Guide for Developing System Requirements Specifications (IEEE1233, 1998) defines requirements as

- A condition or capacity needed by a user to solve a problem or achieve an objective;
- A condition or capability that must be met or processed by a system or system component, to satisfy a contract, standard, specification or any other formally imposed document;
- A documented representation of a condition or capability including the components that is mentioned in the first two definitions.

Another definition was made by (Kotonya & Sommerville, 1997):

- A facility at the user level;
- A general property of the system;
- A specific constraint over the system;
- The specification of a particular algorithm that should be applied to particular calculations;
- A constraint over the development process.

The concept of “requirements” is closely related to the end user or the customer of a particular product. The word “product” is a representative term and does not indicate only a physical issue; a product may be a system, software, service, or a project that is performed with respect to a specified contract. Thus, the result or the success of the project depends on its performance on how many requirements of the customers are met. Thus, an analysis phase should be carried before a design phase to find out the needs of the customers and to convert them into design issues. “End users” or “customers” refer to the people who use the product and create a value from its characteristics. Especially in software engineering terminology, the word “stakeholders” is used to define all people related to the product.

The general definitions do not specify the source of the requirements, i.e., characterizing the events captured by the requirements engineering process and documented as the requirements of the system being developed. Considering this situation, it is difficult to specify “what” without specifying “how”, as the difference between specification and design (Davis, 1993). According to (Jackson, 1995), requirements deal with statements about the application domain, and not about the machine. Describing the requirements of the system refers to describing the phenomena presented in the context, as well as the relationships between them. This idea brings two different thinking about the world (Zave & Jackson, 1995). One describes the reality as it is, without a machine; and the other describes the world as it is desired to be, in presence of the machine. The statements in the first kind of the description are called indicative and in the latter, optative. Optative statements are called requirements (Lopes & Barreto, 2002).

Natural language is the only way to represent requirements so that they can be interpreted by all stakeholders. However, requirements in natural language are too often hard to understand, ambiguous, obscure and misinterpreted (Jackson, 1995). There exist many reasons for this situation; it is difficult to specify complex conditional clauses; terminology is generally imprecise and inconsistent; another difficulty occurs in representing requirements without considering any kind of specific knowledge of any particular stakeholder. Besides the difficulty in understanding the nature of the requirements, there are many other drawbacks in representing them. The use of a language or notation with embedded semantics would surely lower the size of the model. There exist many studies using design notations in the requirements modelling activity (Kotonya & Sommerville, 1996); (Davis, 1993). Considering these difficulties about gathering and representation of requirements, requirements descriptions should provide the following characteristics before the design phase is started:

- Validity: requirements should satisfy any need of any stakeholder
- Reality: requirements should be realistic and be met in the context of the problem (technology, market, corporative)
- Verifiability: there should be a process limited in time and cost in order to check the requirement is met.
- Requirements should be unambiguous (for industry).

The difficulty in gathering and validating the requirements improves its importance within the whole project and forces the analysts and designers to schedule relevant time on requirements analysis. Requirements analysis phase is performed until the target needs of the customers are discovered and validated. Figure 4.1 shows the partition of the requirements analysis phase for any product designed for customers or users.

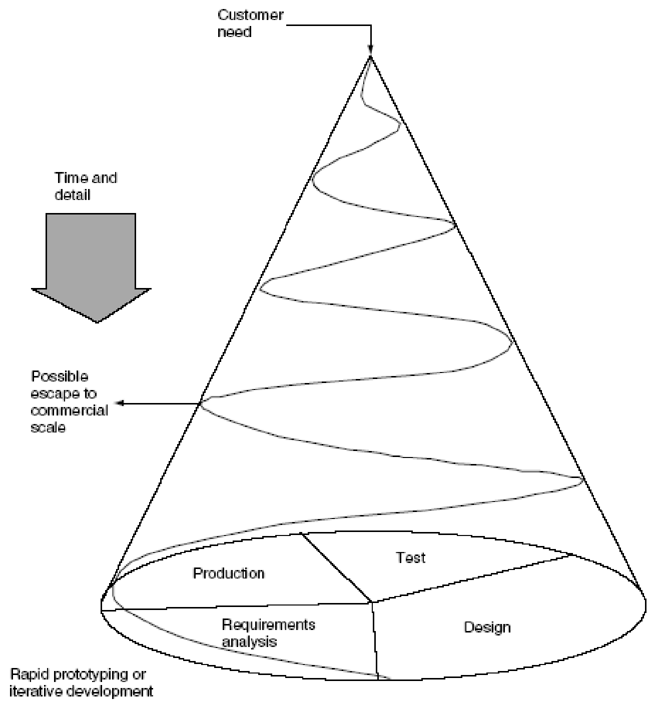


Figure 4.1 The design cycle for a product or system.
Reference: (Grady, 2007, p.31)

Statistics show that nearly 25% of the cost of any project belongs to requirements analysis phase and 45% of it belongs to testing phase. Testing phase is directly affected by how successfully the requirements have been analyzed. These percentages indicate the importance of requirements analysis and modelling phases of projects (Duenas, 2004).

Gathering requirements and analyzing them to define design characteristics are the major task of requirements engineering. Section 3.2 defines these concepts and the following sections explain in details how the requirements analysis and modelling phases are handled.

4.2 Requirements Engineering

"Requirements engineering is the branch of systems engineering concerned with the real-world goals for, services provided by, and constraints on a large and complex software-intensive system. It is also concerned with the relationship of these factors to precise specifications of system behaviour, and to their evolution over time and across system families." (Zave, 1994)

"Requirements engineering focuses on improvements to the front-end of the system development life-cycle. Requirements engineers establish the needs that have given rise to the development process and organising this information in a form that will support system conception and implementation. Engineers/Analysts are asked to note the broad systems engineering remit of requirements engineering" (Finkelstein, 1994).

Requirements are generally analyzed by requirements engineers considering different perspectives, namely, views. Three views are defined in general: functional requirements describing the services or functions of the system with their reaction to behaviour under specific conditions; non-functional requirements describing restrictions or conditions on the operation of the system; and domain requirements specifying in the application domain of the system. These views should be analyzed and designed with respect to the needs of the customers or system users. Thus, requirements are analyzed and modelled until the engineers are sure about meeting the customer needs before design (Finkelstein, 1994).

4.3 Stepwise Requirements Analysis

Requirements analysis does not have the same meaning with system design. Requirements analysis concentrates on the boundaries of the design problem. Analysis is interested in the nature of the enterprise and its characteristics during the use of information whereas the design is the particular application of a specific technology to define the enterprise. Analysis covers what is to be done, not how to do it.

Requirements analysis can be defined as the process of translating business users' views of an enterprise into a designer's view. A business owner's view is usually defined in terms of industrial terminology and is about different mechanisms involved in running an enterprise. Requirements analysis should analyze this situation to determine the fundamental structures and functions of the business, in order to suggest new improvements.

Design projects are handled in accordance with vision and mission of the enterprise. Enterprise management should determine a strategy for new projects. Once a strategy defined and explained in terms of vision and mission, requirements analysis performs the following processes for each project as defined by (Hay, 2003).

1) Scope Definition: Re-examine the scope given to the project by the strategy phase. Does it still make sense? Determine how big the project is. Confirm what data categories will be covered, and what functions. Confirm what part of the organization will be addressed.

2) Plan the Analysis**:** Lay out the steps specifically, identifying who will do each and defining how each can be done successfully.

3) Gather Information**:** Meet with the people who will own the system. They are the ultimate source of all information about the company and what it should do.

4) Describe the Enterprise**:** Use the modelling techniques to portray the six dimensions of the enterprise: what data, how it is processed, where things are done, who plays what roles in the enterprise, when events take place that trigger activities, and why the enterprise is constrained the way it is.

5) Take Inventory of Current Systems**:** While requirements analysis, for the most part, is not concerned with technology or current systems, it is useful to know what exists, and what roles technology plays in the current operation of the enterprise. While the other analysis steps are being done, this is a good time to take stock of current systems and what they are used for.

6) Define What is Required of a New System**:** This is the "requirements" part of requirements analysis. What motivated this project? What specifically would make people's lives and work easier? What kinds of technology look promising?

7) Plan for Transition**:** If a system that is built based on this analysis is at all significant, it will change the infrastructure of the organization. In addition to the mechanics of installing a new system, transition will entail extensive education and training, and it will probably involve organizational changes as well. Planning for this should begin during the requirements analysis phase.

Section 4.3.1 to 4.3.7 explains how these phases should be handled and which activities should be done. These sections are compiled from (Hay, 2003).

4.3.1 Scope Definition

The scope of the strategy study should be the whole enterprise, with definitions also of the scope of each project ("Replace the general ledger system", "Create an e-business", etc.). The definition of that scope, however, may be further refined at the beginning of the requirements analysis project. What is needed now, then, before the requirements analysis process can begin in earnest, is to specify that scope in terms of the Information Architecture's columns:

- Data: What things of significance define the scope of the project?
- Activities: Which activities are to be included in the project?
- Organizations: Who will be involved in the activities?
- Locations: Where will the activities be addressed?
- Timing: Which events are in scope?
- Motivation: Which corporate goals and objectives are being addressed?

The strategy study should have listed the basic things of significance to the business-people, organizations, products, and so on. The scope should now be defined in terms of which of those broad categories are to be addressed. The strategy study should also list, at least in global terms, the functions of the business. The scope statement for a project should at least be articulated in terms of those functions.

4.3.2 Plan the Process

Among other things, the planning process includes identification of the key users and others who will be the source of the analysis information. These are often referred to in the industry as subject matter experts. These are the people who will be interviewed, attend modelling sessions, and so forth. These will be the final arbiters of whether the resulting system performs its intended functions. Ideally, a subject-matter expert should be high enough in the organization to provide perspective, but not so high as to be ignorant of the detailed business processes.

4.3.3 Process Three: Gather Information

So, how can the analysts learn about the enterprise and its requirements? The steps are:

- Step 1: Conduct briefing. Analysts and designers are introduced to the people who will be relying on for information.
- Step 2A: Conduct interviews. Speak to subject-matter experts individually to learn the nature of their work.
- Step 2B: Conduct "joint application development" (JAD) sessions. Alternatively, speak to people in small groups, developing models with their assistance.
- Step 3: Obtain industry information and patterns. Seek out information about how other companies in this industry work. A similar project is very likely to have been done before in another company. Take advantage of that, if possible.
- Step 4: Review the range of available software. At all costs designers should avoid having the characteristics of available software lead the analysis, but it is sometimes possible to learn important things about the nature of the business from the design of software that has served similar functions.

The primary deliverable from this process is a set of notes and sketches, along with a log of who was seen. It also includes any supplemental industry and current systems information that might be available as well as samples of as many reports and forms as can be collected. There exist additional approaches to gather the information from the users depending on analytical techniques. One of these approaches is proposed for enterprise modelling in chapter 6.

4.3.4 Process Four: Describe the Enterprise

In requirements analysis models are needed that describe the business the way the business owners see it and models that describe it in more architectural terms. Briefly, the categories of modelling effort to be done are the following:

- Step 1: Create data (object class) models— Identify things of significance about which the organization intends to collect information.
- Step 2: Create activity models— Identify both the current processes and the underlying functions of the organization.
- Step 3: Create location models— Identify where the business is conducted.
- Step 4: Create people and organization models— Identify who plays what roles in the operation of the business.
- Step 5: Create event and timing models— Identify how time affects the operation, in terms both of corporate schedules and of the events that cause things to happen in the company.
- Step 6: Create motivation models— Identify the business policies of the enterprise, as well as the strategies and tactics they support and the business policies and rules derived from them that constrain the way the business works.
- Step 7: Present models— Show the models to as many business-area experts as possible, obtaining corrections and enhancements, along with agreement on the final product.

4.3.5 Define What Is Required of a New System

The "requirements" of requirements analysis are the specification of what is to be done with a new system. This does not mean that a new system itself is to be described in detail. Rather, what is produced is a statement that includes:

- The purpose of a proposed system
- Key players
- Required capabilities
- Requirement constraints
- Non-functional requirements
- The level of technology to be employed
- Capacity requirements
- The decision to make or buy the new system

Step 1: Restate Project Purpose

In the beginning, there was a reason why this requirements analysis project was initiated. There was a perceived need for information or processing that was clearly not being met by current systems. In all cases, there is an overriding business need, expressed in the strategy report that created the project in the first place. Write that down, and publish it as the frontispiece to all other project documents.

Step 2: Identify Key Players

The ultimate success of any project will be its acceptance by the people who will depend on it for their jobs. These people are the ultimate source of all requirements definitions and defined as follows (Robertson & Robertson, 1999):

- Clients and customers: A client of a project is one who pays for the development of the product. This person has financial responsibility for it until it is delivered. A customer is a person who will pay for the final product. You must understand these people well enough to build a product they will buy and use.

- Users: A user is a person who will ultimately work with the product. Satisfying a user comes from designing a product that can be operated effectively. This requires designers to have considerable understanding of the work to be done.
- Stakeholders and consultants: Robertson & Robertson (1999) describe stakeholders as "people who have an interest in the product. They will manage it, they will use it, or they will in some way be affected by its use. Stakeholders are people who have some demands on the product, and hence must be consulted in the requirement gathering activity". These include management, business subject-matter experts, safety inspectors, the Legal Department, and others. They may also include outside groups, such as the marketplace, professional bodies, special interests, and cultural interests. A consultant (internal) typically does not have a vested interest in the project but may know useful things about the enterprise or about this application.
- Information-technology workers: These people typically should not be the source of requirements information, but they should be active participants in the analysis process, so that they know when the requirements come.

Step 3: Identify Required Capabilities

Robertson & Robertson (1999) describes functional requirements as "the things the product [new system] should do—an action that the product must take if it is to provide useful functionality for its user. Functional requirements arise from the fundamental reason for the product's existence".

In addition, non-functional requirements are "properties, or qualities, that the product must have. In some cases the non-functional requirements are critical to the product's success".

To arrive at these required capabilities, then, it is necessary not to ask the users what they want, but to look at the models, examining the difference between the business owners' views of their current systems and the architect's view. What data that are in the enterprise model are missing from the current world? What processes could be rendered more rational?

Following tasks should be performed to answer these questions:

1) Task 1: Identify Missing Data
2) Task 2: Identify Missing Functions
3) Task 3: Propose Systems and Define Use Cases for Them

Once a system is verified, designers can use use-cases to describe exactly how the system will behave (Jacobson, 1992). The graphic for a use case is simply one or more stick figures, representing the actors, plus an ellipse representing a system being interacted with. A use case does not document in detail the data flowing into and out of the system, but it does represent in its underlying documentation the interactions between the actors and the potential system. A use case is typically documented in terms of its purpose, plus the set of triggers issued by the actors and its responses to each. The set of responses is often documented in terms both of the normal responses if all goes well, and of alternative responses if something does not (Cockburn, 2000).

Step 4: Identify Requirement Constraints

A requirement constraint limits the design choices available to meet one or more required capabilities. That is, an engineer may want to manage inventory, but there are requirement constraints that limit how analysts and designers can go about doing so. These include hardware platforms available, budgetary limits, and architectural decisions previously made.

Specifically, these may include input and output constraints having to do with restrictions in the environment about how data can be entered. In a manufacturing environment, for example, terminals may have to be hardened and made immune to dust and chemicals.

Also included are other design constraints, which derive from economics, existing systems, and training constraints. For example, certain technologies may be prohibitively expensive, there may be restrictions as to what data are available from feeding systems, or there may be specific requirements for the user interface.

Step 5: Identify Non-functional Requirements

A non-functional requirement is a property or quality that the proposed system must have to support the functional requirements. These include such things as:

- Quality
- Response time
- Look and feel
- Security
- Cultural
- Legal

The tasks defined in this phase can be represented as follows:

1) Task 1: Identify Quality Requirements. It is not sufficient to say simply that a new system will calculate an account balance.
2) Task 2: Define Response-Time Requirements
3) Task 3: Define Look and Feel Requirements
4) Task 4: Define Security Requirements
5) Task 5: Define Cultural and Political Requirements
6) Task 6: Define Legal Requirements

Step 6: Determine Level of Technology

The requirements analysis phase should be carried out independent of technology. The assignment here is to determine what data and processing a business requires to carry out its objectives. Once these requirements have been stated, then the design phase can apply technology to these requirements.

It is appropriate, however, at the end of the requirements analysis phase to indicate desirable technological directions to take. Steps 6, 7, and 8 of Process Five, as well as Process Six, below, begin the process of addressing the technology that ultimately will be used to implement any new system. These steps could as easily be considered at the beginning of design as at the end of requirements analysis. The point here is that they constitute the transition from one to the other.

Step 7: Identify Capacity Requirements

As mentioned above, the activity and data models should include measures of the "size" of each. In the case of the data model, this means, for each entity type, the number of occurrences expected. This includes the number expected initially as well as the number expected over (for example) the next five years, with a projected growth rate. In the case of the function and process models, this means a measure of how often the activity is carried out, along with some measure of its relative complexity. This information can then be used by the designer to estimate disk space and processing requirements.

Step 8: Decide Whether to Make or Buy

Requirements analysis is just as important if software is bought as it is if it is built by the designer. Designers cannot adequately evaluate software unless they have a clear idea of what it should do and what the underlying structure of its data is.

Note that if the functions to be automated are routine maintenance functions, like accounting, which are not central to the business, it is perfectly appropriate to use standard, commercial software to address them. On the other hand, if company engineers are automating a part of the business which is central to the operation—which is at the heart of what makes managers stand out from competition—then it can be assumed that commercial software has not addressed the points that are unique to the company. In this case, it is better off developing the application.

Step 9: Deliverable: Requirements Statement

The deliverable from Process Six, then, is a report itemizing

- The project goals
- Key players
- Functionally required capabilities

- Non-functional requirements
- Required constraints
- Level of technology
- Capacity requirements
- A discussion of the decision whether to build a new system or to buy one

4.3.6 Process Six: Determine the Existing Systems Environment

As stated above, the set of current systems should not define the requirements for systems in an organization. The requirements analysis phase should be carried out independent of technology. The assignment here is to determine what data and processing a business requires to carry out its objectives. Once these requirements have been stated, then the design phase can apply technology to these requirements.

Still, to the extent that systems usually constitute an important part of the business owners' views of their current environment, it is useful to know just what exists presently and the roles of various systems in the way the enterprise's business is carried out. Moreover, it is important to document the existing systems environment. This knowledge can provide useful insights in preparing the models described above, and it is important during transition when the time comes to move from the existing systems environment to a new one.

Because the skills required for conducting this kind of research are different from those used to model the business, this process, determining the existing systems environment, can be done by a separate team in parallel with and at the same time as other processes in the requirements analysis project. This process is one of making sure designers understand not just what systems exist and what they do, but also the operating environment,

- The physical architecture,
- The technical architecture,
- Operating procedures, and
- Capacity.

Specifically, it involves the following steps:

Step 1: Define Operating Environment

Step 2: Identify Software Environment

Step 3: Define Technological Architecture

Step 4: Define Operational Procedures

Step 5: Identify Existing Capacity

Step 6: Deliverable: System Inventory

The steps in this process will result in reports and lists of various kinds. These will provide the basis for planning further development phases.

4.3.7 Process Seven: Plan for Transition

Looking at the system development life cycle, the most problematic of all the phases is transition. Transition is the establishment of a new system as part of the infrastructure of the enterprise. It involves education, training, implementation of software, and conversion of data. It addresses the conversion of a set of existing business owners' views to a new set. This recognizes that, if this system is at all innovative, it will require the enterprise to change the way it does business. If this project does not provide a new tool that will be different from previous tools, designers and/or analysts have to ask whether it is worth doing. To do it means that many in the enterprise will have to change the way they do their jobs. Indeed, many may have different jobs altogether.

Step 1: Begin Reorganization

Step 2: Begin Education

Step 3: Prepare for Training

Step 4: Prepare for Data Conversion

Step 5: Prepare for Implementation of Hardware and Software

Step 6: Deliverable: Transition Plan

The requirements analysis phase is not too early to begin consideration of the hardware and software implementation process. Included in many of the modelling techniques, but especially in entity/relationship modelling, is provision for estimating how many things there will be. How many products? How many contracts?

If these requirements are significantly greater than what is currently available, the time is ripe for beginning the process of acquiring the additional capacity. This may mean buying new computers or simply expanding the amount of disk space available. Or, if the architecture is changing (as in the move from a client/server architecture to the World Wide Web), major capital expenditures may be required.

An enterprise may already have procedures for implementing new software, but if not, consideration must be given to testing and the migration of software from "test mode" to "production mode". What will be the exact steps, and how will they be controlled?

4.4 Analysis Perspectives in Requirements

This section explains how the requirement analysis process is handled and from which perspective they are carried out. Its structure and subsections are compiled from (Grady, 2007, p.39-55).

4.4.1 Functional Analysis

This process starts with the need as function F, which is expanded into a set of next-tier functions, which are all things that have to happen in a prescribed sequence (serial, parallel, or some combination) to result in function F being accomplished. One draws a block for each lower-tier activity and links them together in a sequence using directed line segments to show a sequence. Logical OR and AND symbols are used on the connecting lines to indicate combinatorial possibilities that must be respected. This process continues to expand each function, represented by a block, into lower-tier functions. Figure 4.2 sketches this overall process for discussion. A function statement begins with an action verb that acts on a noun term. The functions exposed in this process are expanded into performance requirements statements that numerically define how well the function must be performed. This step can be accomplished before or after the allocation of the performance.

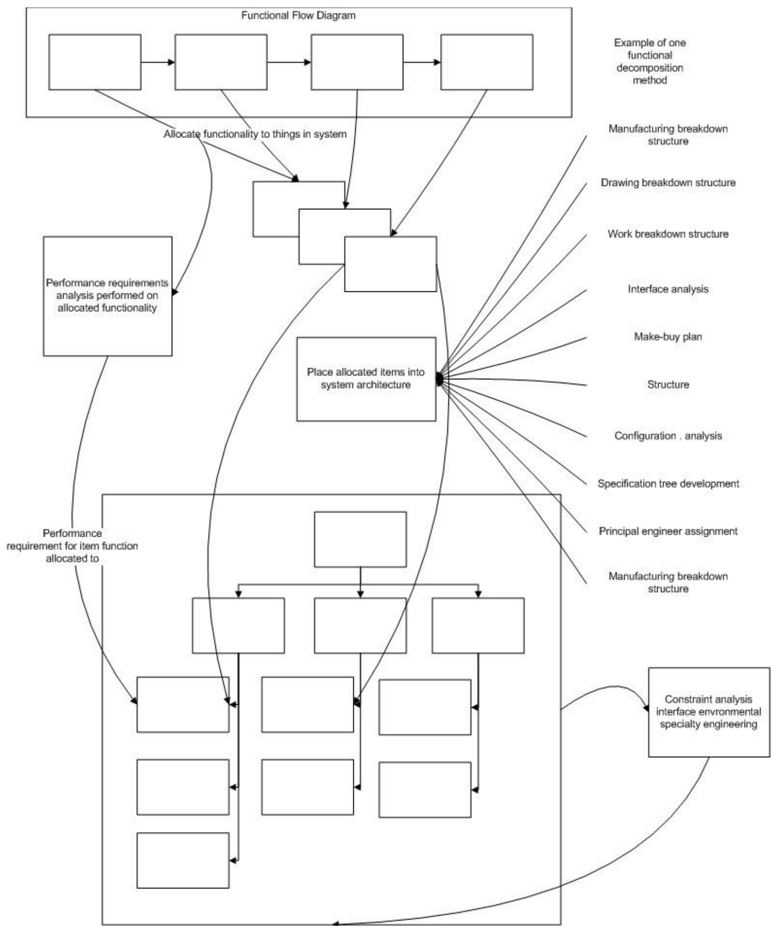


Figure 4.2 Decomposition of systems and hardware.
Reference: (Grady, 2007,p.50)

But, in the preferred case, the identification of the function obligates the analyst to write one or more performance requirements derived from the function and allocate that performance requirement to an entity to which it is allocated. This is the reason for the power of all decomposition techniques. They are exhaustively complete when done well by experienced practitioners. It is less likely that designers will have missed anything compared to an ad hoc approach. This process begins with the need and ends when the lowest tier of all items in the physical product entity structure in each branch satisfies one of these criteria: (1) the item will be purchased from another company at that level or (2) the developing organization has confidence that it will surrender to detailed design by a small team within the company and that the corresponding problem is sufficiently understood in either case that an adequate specification can be prepared.

4.4.2 Performance Requirements Analysis

Performance requirements define what the system or item must do and how well it must do those things. Precursors of performance requirements take the form of function statements or functional requirements (quantified function statements). These should be determined as a result of a functional analysis process that decomposes the customer need as noted above using an appropriate flow diagramming technique. Many organizations find that they fail to develop the requirements needed by the design community in a timely way. They keep repeating the same cycle on each program and fail to understand their problem. This cycle consists of receipt of the customer's requirements or approval of their requirements in a specification created by the contractor, followed by a phony war on requirements where the systems people revert to documentation specialists and the design community creates a drawing release schedule in response to management demand for progress. As the design becomes firm, the design people prepare an in-house requirements document that essentially characterizes the pre-existing design. Performance requirements are traceable to (and thus flow from) the process from which they are exposed much more effectively than in a vertical sense through the product entity structure.

4.4.3 Design Constraints Analysis

Design constraints are boundary conditions within which the designer must remain while satisfying performance requirements. All of them can be grouped into the three kinds, described below. Performance requirements can be defined prior to the identification of the things to which they are ultimately allocated. Design constraints generally must be defined subsequent to the definition of the item to which they apply. Performance requirements provide the bridge between the problem and solution planes through allocation. Once the product entity structure is established, three kinds of constraints analysis can be applied to these items.

4.4.4 Interface Requirements Analysis

Systems consist of things. These things must interact in some way to achieve the need. A collection of things that do not in some way interact is a simple collection of things, not a system. An interface is a relationship between two things in a system. This relationship may be completed through many different media, such as wires, plumbing, a mechanical linkage, or a physical bolt pattern. These interfaces are also characterized by a source and a destination—that is, two terminals, each of which is associated with one thing in the system. Developing systems is constructed to identify the existence of interfaces and then to characterize them, each with a set of requirements mutually agreed upon by those responsible for the two terminals. Note the unique difference between the requirements for things in the system and interfaces. The things in systems can be clearly assigned to a single person or team for development (Grady, 2007).

4.4.5 Environmental Requirements Analysis

One of the most fundamental questions in system development involves the system boundary. Designers should be able to unequivocally determine whether any particular item is in the system or not in the system. If it is not in the system, it is in the system environment. If an item is in the system environment, it is either important to the system or not. If it is not, they may disregard it in an effort to simplify the system development. If it is important to the system, designers must define the relationship to the system as an environmental influence. Designers may categorize all system environmental influences in the five following classes:

1. *Natural environment*—Space, time, and the natural elements such as atmospheric pressure, temperature, and so forth. This environment is, of course, a function of the locale and can be very different from that with which designers are familiar in our immediate surroundings on Earth, as in the case of Mars or the Moon.

2. *Hostile systems environment*—Systems under the control of others that are operated specifically to counter, degrade, or destroy the system under consideration.

3. *Noncooperative environment*—Systems that are not operated for the purpose of degrading the system under consideration but have that effect unintentionally.

4. *Cooperative systems environment*—Systems not part of the system under consideration that interact in some planned way. Generally, these influences are actually addressed as interfaces between the systems rather than environmental conditions because there is a person from the other system with whom designers may cooperate to control the influences.

5. *Induced environment*—Composed of influences that would not exist but for the presence of the system. These influences are commonly initiated by energy sources within the system that interact with the natural environment to produce new environmental effects. As noted above, cooperative environmental influences can be more successfully treated as system interfaces. Hostile and non-cooperative influences can be characterized through the identification of threats to system success and the results joined with the natural environmental effects.

The challenge to the system engineer is to isolate on those parameters that are important and those that are not, and then to select parameter ranges that are reasonable for those parameters that will have an impact on our system under development. The union of the results of all of these analyses form the system environmental requirements. It is not adequate to stop at this point in the analysis.

4.4.6 Specialty Engineering Requirements Analysis

The evolution of the systems approach to development of systems to solve complex problems has its roots in the specialization of the engineering field into a wide range of very specialized disciplines for the very good reasons noted earlier.

The challenge in system engineering is to weld these many specialists together into the equivalent of one all-knowing mind and applying that knowledge base effectively to the definition of appropriate requirements, followed by development of responsive and compliant designs and assessment of those designs for compliance with the requirements as part of the verification activity.

Specialty engineers apply two general methods in their requirements analysis efforts. Some of these disciplines use mathematical models of the system, as in reliability and maintainability models of failure rates and remove-and- replace or total repair time. The values in these system-level models are extracted from the model into item specifications. Commonly these models are built in three layers. First, the system value is allocated to progressively lower levels to establish design goals. Next, the specialty engineers assess the design against the allocations and establish predictions. Finally, the specialists establish actual values based on testing results and customer field use of the product. Another technique applied is an appeal to authority in the form of customer-defined standards and specifications. A requirement using this technique will typically call for a particular parameter to be in accordance with the standard. One of these standards may include a hundred requirements, and they all flow into the program specification through reference to the document unless it is tailored. Specialty engineers must, therefore, be thoroughly knowledgeable about the content of these standards; familiar with their company's product line, development processes, and customer application of that product; and knowledgeable about the basis for tailoring standards for equivalence to the company processes and preferred design techniques.

Section 4.4 presented the perspectives and views which are considered in the requirements analysis studies. Functionality, performance, interface design, environmental issues and constraints are different perspectives (or dimensions) of a software project, thus the requirements analyses are affected by these perspectives. During the analysis engineers/analysts or designers collect data from interviews and observations and organize them in forms, diagrams, and charts. Section 4.5 summarizes these documents and techniques that support the analysis studies.

4.5 The Documents and Techniques Used for Requirements Analysis

4.5.1 Documents

Requirements are necessary attributes defined for an item prior to efforts to develop a design for the item. System requirements analysis is a structured, or organized, methodology for identifying an appropriate set of resources to satisfy a system need and the requirements for those resources that provide a sound basis for the design or selection of those resources. It acts as a transformation between the customer's system need and the design concept energized by the organized application of engineering talent. The basic process decomposes a statement of customer need through a systematic exposition of what the system must do to satisfy that need. The need is the ultimate system requirement from which all other requirements and the designs flow (Grady, 2006, p.30).

Allocation of functionality results in identification of things that have to be fitted into the physical model of the system, i.e.,, the architecture. The principal outputs include

1) Requirements analysis sheet. The RAS, in paper or computer screen format, captures the relationships between the functions, the derived entity capability (performance) requirements, and the product architecture entities that accomplish the functionality.

2) System description document, captures the system diagrams defining system composition. It includes (a) a functional flow diagram, illustrating needed system functionality; (b) the aggregate RAS (in full or by reference to a computer database tool containing it), capturing the allocations of functionality to architecture; (c) an architecture block diagram, a hierarchical block diagram, defining the things in the system; (d) a schematic block diagram, defining the relationships between all the things shown on the architecture block diagram; and (e) any other documentation products of the system definition process.

3) Drawing breakdown structure. The engineering drawing overlay of the product architecture, telling what engineering drawings will be produced.

4) Specification tree. The specification overlay of the product architecture, telling what specifications will be prepared and what format they will follow.

5) Manufacturing breakdown structure. The manufacturing overlay of the product architecture, defining the groups of things moving from one major production area to another.

6) Configuration/end item list. Identification of the things through which the program will be managed.

7) Integrated Product/Process Team (IPPT) responsibilities. Team responsibility boundaries relative to the architecture.

8) Work breakdown structure. The product component of the WBS is an infrastructure of the product architecture, upon which the whole program plan is based (see Figure 4.2).

4.5.2 Requirements Statement

Requirements analysis and requirements writing are necessary but difficult, sometimes tedious, tasks. In this chapter four strategies are explained—one or more of which, hopefully, will reduce the requirements analysis difficulty by providing mechanisms for gaining insight into the attributes that should be controlled by written requirements statements. The fact is that it is not hard to write requirements. It is hard to know what to write them about and to determine appropriate numerical values to include within them.

A normal requirement statement is a written statement of a requirement in one or more complete sentences in a language familiar to the customer (normally English in the U.S.) using the idiom of the particular business sector (aerospace for example). Good sense and common specification standards require that the content of a specification include complete sentences organized in a particular way. Each requirement statement in a specification must satisfy several characteristics including: (1) proper grammar, (2) appropriate use of shall, will, and other key words, and (3) rigid compliance with a format. But the need, during early contract phases, to immediately prepare complete requirements statements satisfying a specification style guide can act as a barrier to the early, timely identification of needed technical requirements as a prerequisite to early design concept development and procurement work.

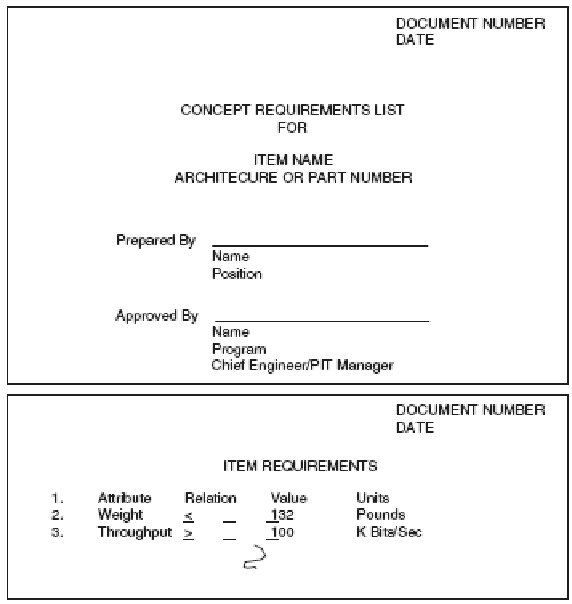
In an effort to unburden early concept development work of unnecessary rigor, reduce the cost of such work, and improve the early requirements identification capability on a project, a Concept Requirements List (CRL) is recommended. This document is composed of nothing but a cover and a numbered list of primitive requirements statements.

The structure of a CRL is very simple. In *paper form* it consists of a cover page and as many pages of requirements as needed. Each page contains primitive requirements statements. These statements are numbered from 1 through *n*. No effort is made to format the primitive statements by category (performance or constraints) or to follow a special prescribed sequence. The statements may simply be listed in the order conceived by the element principal engineer. The document may be published either in book form or presentation form. A sample CRL is presented in Figure 4.3.

1) *Architecture Block Diagram (ABD).* An architecture block diagram is a hierarchical diagram consisting of simple blocks depicting the elements of a system illustrating the family structure of the system. If an ABD is included in the CRL, it should consist of one block for the subject block and one block for each of the immediately subordinate elements arranged below it and interconnected by a series of lines denoting hierarchy. The primitive statement would be, "Element architecture as defined in Figure ...."

2) *Schematic Block Diagram (SBD).* A schematic block diagram illustrates the interfaces required in a system or element thereof by connecting blocks from the architecture block diagram with lines indicating an interface requirement between a

pair of elements. A schematic block diagram illustrates the interfaces between the element that is the subject of the document and all other system elements. This diagram consists of a block titled with the name of the element and one block for each unique element the subject element interfaces with. One or more lines are drawn between the subject block and each interfacing block. Alternatively, an *n-square* diagram could be used.



DOCUMENT NUMBER
DATE

CONCEPT REQUIREMENTS LIST
FOR

ITEM NAME
ARCHITECURE OR PART NUMBER

Prepared By ____________
Name
Position

Approved By ____________
Name
Program
Chief Engineer/PIT Manager

DOCUMENT NUMBER
DATE

ITEM REQUIREMENTS

1.	Attribute	Relation	Value	Units
2.	Weight	≤	132	Pounds
3.	Throughput	≥	100	K Bits/Sec

Figure 4.3 Concept requirements list structure.
Reference: (Grady, 2006, p.49)

3) *Functional Flow Diagram (FFD).* A FFD may be included to illustrate the identification and sequence of functions the element must satisfy as a precursor of performance requirements captured in other primitive requirement statements. The referencing primitive requirement statement would be “Element functions as defined in Figure ...”

4) *Process Flow Diagram (PFD).* As an alternative to a FFD where the elements of the system are already well established and functional analysis is not necessary as a structured system decomposition tool, a process flow diagram may be included. The PFD is similar to the FFD but it is created as an analogy of the planned real world operating system process. A functional flow diagram is frequently prepared

when the exact composition of the system is not yet known. The primitive statement would be similar to that shown for the FFD.

5) *Timeline Diagram (TLD*). A timeline may be included that defines critical timing of signals or events that the element design must respect. It should consist of a Gantt-type chart where the bars are defined on a vertical axis and their length indicates time duration against a horizontal time scale. If an FFD is also included, the FFD blocks and the timeline blocks should be coded with corresponding numbering.

4.5.3 Requirements Derivation

Many design engineers like to separate all of the requirements that pertain to the item for which they are responsible into two sets: (1) source or customer requirements, and (2) derived requirements. This may be a useful distinction for an engineer working on what he or she perceives as an isolated development task. The system engineer should realize that in the development of an unprecedented systems, every requirement but one, the system need, is derived from the need. In the overall requirements analysis process, the derived distinction tends to have little special significance since all requirements except the ultimate customer needs are derived.

4.6 Requirements Modelling

A requirements model especially handled in software engineering studies is expected to contain an overall description of functions; any people, physical things, concepts and the interactions among them that are important to the engineer's understanding of application domain, and business situations in details to evaluate possible designs. Requirements should be organized in a model which will be useful for designing the software.

Requirements modelling deals with the development of requirements covering

- iterative and cooperative processes for analysis of the problem;
- goals, functions, and restrictions of the system;
- documentation of the results;
- checking the result to be complete, consistent, and relevant.

The core of requirements engineering, and the primary means by which the needs are rendered in a form that can be used to realise them, is the identification of the goals that a projected system is required to satisfy and the services that it should supply. The goals may have interdependencies or conflicts which must be modelled and where appropriate resolved. In certain circumstances, goals that may be interpreted as service can be used to predict the problems they might encounter provision; however, in identifying these it is necessary to suggest ways in which the interface to the system identifies the "external" actions the system should perform. A number of approaches have emerged which explicitly represent goals and build a system model round these goals. Though these approaches differ in specifics, the broad outline is the same.

Requirements models are generally built based on class diagrams. Class diagrams describe a group of objects with similar attributes, common behavioural characteristics within the operations, common relationships to the other objects, and common semantics. Classes are retrieved from the statements defined in use cases. Nouns and noun phrases are found out to build classes. The phrases are only retained if they help to explain the nature or structure of the application domain. Thus, redundant classes are eliminated from requirements model if they are beyond the system. Classes and objects do not exist in isolation from one another; relationships are defined among them (Jacobson, 1992).

ISO 9126 standardizes the major characteristics which should be covered by a requirements modelling process. These characteristics and the properties covered by them are;

- Function: accuracy, interaction, regulations, security
- Reliability: maturity, recovery.
- Usability: understandability, ease of operation, learnability.
- Efficiency: performance, resource usage.
- Maintainability: analysability, change tolerance, variability, verifiability.
- Portability: adaptability, ease of installation, substitution, conformity.

The requirements modelling process is handled and managed by requirements engineers. Requirement engineering is a process of study of the needs of users in order to get a definition of the hardware-software system. Requirements engineers carry out common activities: elicitation, analysis, specification, verification, validation, and management. Elicitation is the process of discovery, formulation, documentation, and understanding of needs of the user and system restrictions. Analysis is the process of refinement of the needs of the user. Specification is the process of documentation of these needs in a clear, unambiguous, and precise manner. Management is a general activity covering the processes of planning, coordination, and documentation of former activities, evolution and maintenance for requirements.

Requirements models should be traceable so that it can be managed and improved. Traceability is a capability to establish the relations between requirements and other development products including pre-traceability (forward) and post-traceability (backward). Pre-traceability covers describing and following the lifecycle of any requirement before it is included in a formal specification whereas post-traceability covers describing and following the lifecycle of the requirement after it has been included in a formal specification (specification, design, verification, validation). This area has recently seen an upsurge in research interest. The bulk of the work concentrates on the ability to link fragments of text and to visualise navigate these links (Finkelstein, 1994)..

Probably the most difficult task in requirements engineering is information gathering - that is gathering information on the needs and the "domain" or "environment" in which these needs are situated. This information may be set down in large documents, may be held by identifiable experts, and may be buried in the work practices of individual users, and so on. For the most part the techniques available in this area have been borrowed from related fields. Requirements engineering has yet to evolve a distinct set of techniques of its own. The use of structured interviews and questionnaires is frequently cited but little analysed. Similarly text and document analysis techniques such as repertory grids have been drawn from area of knowledge acquisition. An interesting emergent area is the use of ethnographic and associated "observational" methods. It is already evident that any realistic domain requires a judicious selection and combination of techniques. How to make such a selection and combination is however far from clear. There is clearly significant scope for further work in this area.

4.7 UML and USE CASE:

4.7.1 UML: Unified Modelling Language

Generally requirement engineers and requirement models aim at determining the user requirements for a software and system. There exist many techniques and tools improved for requirements modelling, but for the viewpoint of enterprise engineers, the most common techniques are use cases and unified modelling language. Thus, they are considered for the context of this study.

UML (Unified Modelling Language) is another tool to model and analyze requirements of system users. The semiformal and formal languages are not adequate to represent the requirements; because of that UML has common usage. UML is organized according to the different perspectives perceived by the different stakeholders involved with the software development (Jain, Mohan, & Dholakia, 2004). It proposes an architecture consisting of five views:

- ❖ Design view
- ❖ Implementation view

- ❖ Process view
- ❖ Deployment view
- ❖ Use Case view

These views are constructed in hierarchical manner, thus each of them defined and explained correctly in order to supply the right information for the next phase. The overview of UML structure is represented in Figure 4.4.

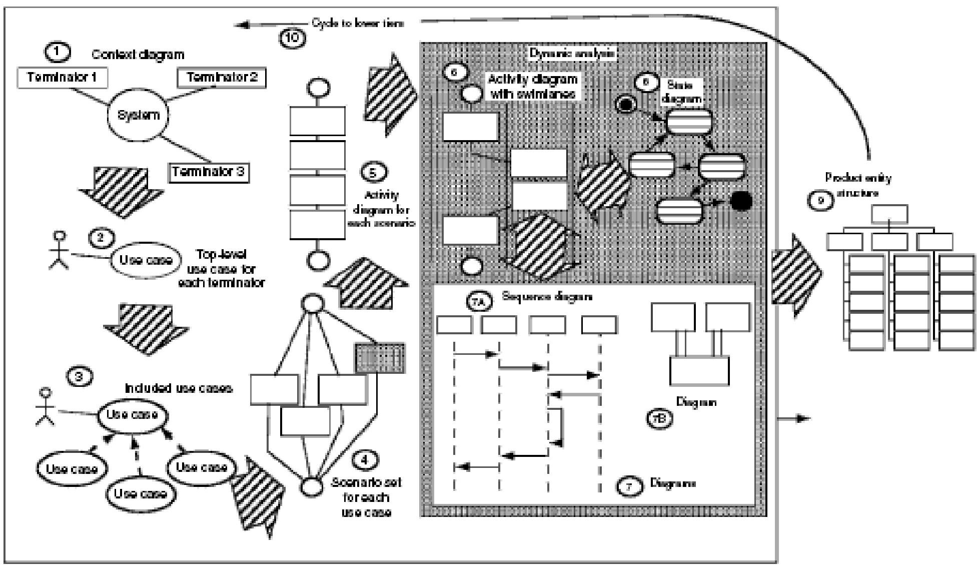


Figure 4.4 UML overview.
Reference: (Grady, 2007, p.60)

Besides that, the language offers a different set of diagrams that are used across the views to represent the static and dynamic aspects of the system, *Use case diagrams* show use cases, actors and the relationship among them. A use case is a sequence of actions that an actor (usually a person, but perhaps an external entity, such as another system) performs within a system to achieve a particular goal (Rossenberg & Kendall, 1999). *Interaction Diagrams* encompass two semantic equivalent diagrams: *sequence diagrams* and *collaboration diagrams.* They both focus on dynamic aspects of the application, the former emphasizing the time ordering of messages and the latter emphasizing the structural organization of the objects that exchange messages. *Statechart diagrams* show a state machine consisting of states, transitions, events and activities. *Statechart diagrams* are used to illustrate the dynamic view of a system, focusing on the event-ordered behaviour of a software artifact (object, system, interface, etc.). *Activity diagrams* show the flow of control among the activities supported by a system (Booch, Rumbaugh, & Jacobson, 1998). Figure 4.5 shows the general structure of an activity diagram and a use case.

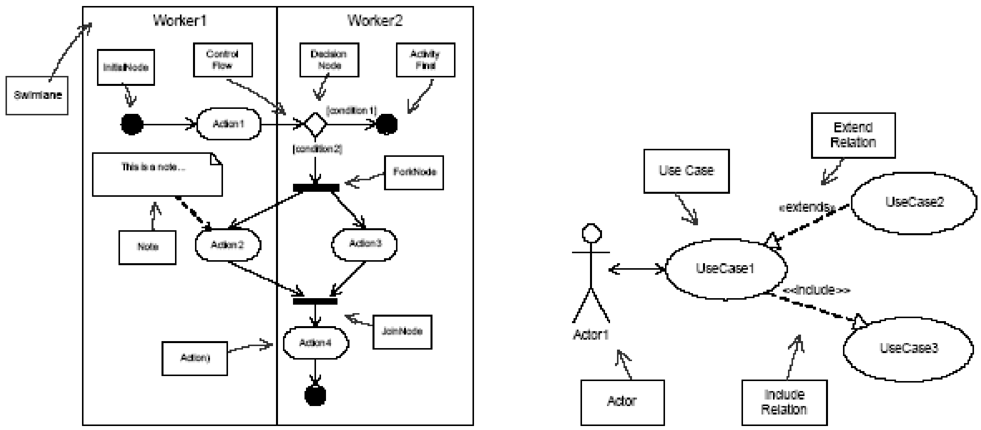


Figure 4.5 Activity and use case diagram notation.
Reference: (Stolfa & Vondrak, 2004)

Class diagrams are the most commonly used diagrams when modelling object-oriented systems. They include classes, interfaces and the relationships among them and are used to show the static aspect of a system. *Object Diagrams* show a set of objects and their relationships. Less usual than *class diagrams,* they are used to depict a snapshot on data structures and on instances of the elements found in the *class diagrams. Component diagrams* show a set of components and their relationships. They are used to illustrate the static organization of the physical system. *Deployment diagrams* show the nodes and relationships among them, representing the physical hardware on which the system is to be installed (Booch, Rumbaugh, & Jacobson, 1998).

Deployment diagrams are related to the *component diagrams* and together they specify how a system is to be deployed. Each node of the *deployment diagram* typically encloses one or more components, described by the *component diagram.*

Table 4.1 summarizes the views and diagrams to model the related components in UML.

Table 4.1 The summary views and corresponding diagrams of UML

Architectural View	Aspect	Diagram
Use Case View	Static	Use case diagram
	Dynamic	Interaction diagrams State-chart diagrams Activity Diagrams
Design View	Static	Class diagrams Object diagrams
	Dynamic	Interaction diagrams State-chart diagrams Activity diagrams
Implementation View	Static	Component diagrams
	Dynamic	Interaction diagrams Statechart diagrams Activity diagrams
Deployment View	Static	Deployment diagrams
	Dynamic	Interaction diagrams Statechart diagrams Activity diagrams
Process View (focus on classes that control threads and process)	Static	Class diagrams Object diagrams
	Dynamic	Interaction diagrams Statechart diagrams Activity diagrams

Reference: (Lopes & Barreto, 2002)

4.7.2 Use Cases

The Use Case Model is a model of what the system is supposed to do and the system environment. The use case model considers requirements modelling that aims to delimit the system and define the functionality that the system.. In the use case approach, the requirements model can be regarded as formulating the functional requirement specification based on the needs of the system users (Sommerville & Sawyer, 1997).

The use case model specifies the functionality the system has to offer from a user's perspective and designers define what should take place inside the system. This model uses **actors** to represent roles the users can play, and **use cases** to represent what the users should be able to do with the system (Adolph, Bramble, Cockburn, & Pols, 2002). Each use case is a complete course of events in the system, seen from a user's perspective. If appropriate, interface descriptions may also be developed. These will specify in detail what the user interface will look like when the use cases are performed. To give a conceptual picture and a better understanding of the system, designers use objects that represent occurrences in the problem domain. This model will serve as a common foundation for all the people involved in the requirements analysis, developers as well as orderers (Jacobson, 1992).

Processes allocate requirements to use cases. A use case is a description of a set of sequences of actions that a system is able to perform in order to produce an observable result for an actor and documented by textual templates developed using UML diagrams. Processes also describe the domain of operation in parallel with use cases. The elements in the domain are represented by concepts, responsibilities, and collaborations. Domain modelling allows for a smoother development. The goal of the requirements model is to describe what the system should do by specifying its functionality. Requirements modelling allows to the developer and the customer to agree with that description. For example, use case models examine the system functionality from the perspective of actors and use cases (Scneider & Winters, 2001). An actor is someone (user) or something (other system) that may interact with the system being developed. A use case is a pattern of behaviour the system exhibits. Each use case is a sequence of related transactions performed by the actor and the system in a dialog. Use case models are described by UML use case diagrams (Stolfa & Vondrak, 2004).

Several models are needed to fully describe the evolving system. The models are developed incrementally across iterations and these iterations are started with use case modelling in both business modelling phase and requirements specifications phase Figure 4.6 shows the contribution of use case models during the design and implementation of a business software project.

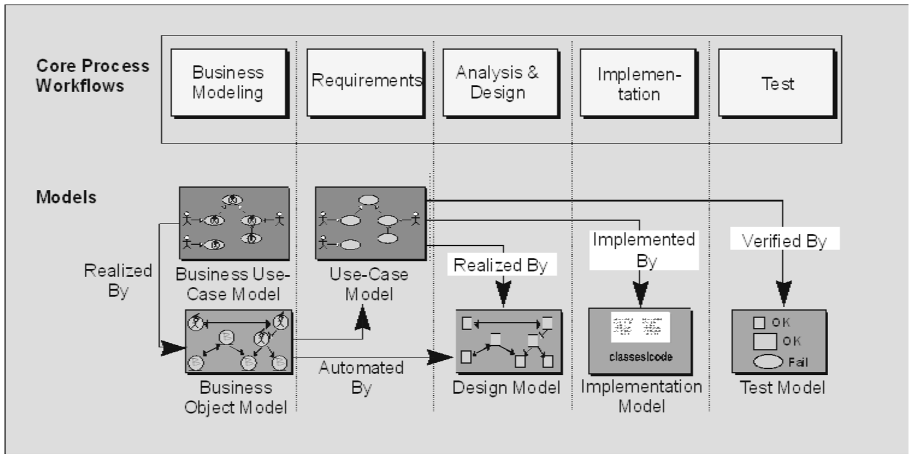


Figure 4.6 Example work flow of the rational unified process .
Reference: (Kruchten, 2000)

To show the requirements definitions using a use case model, Figure 4.7 is presented as an example for ATM system design.

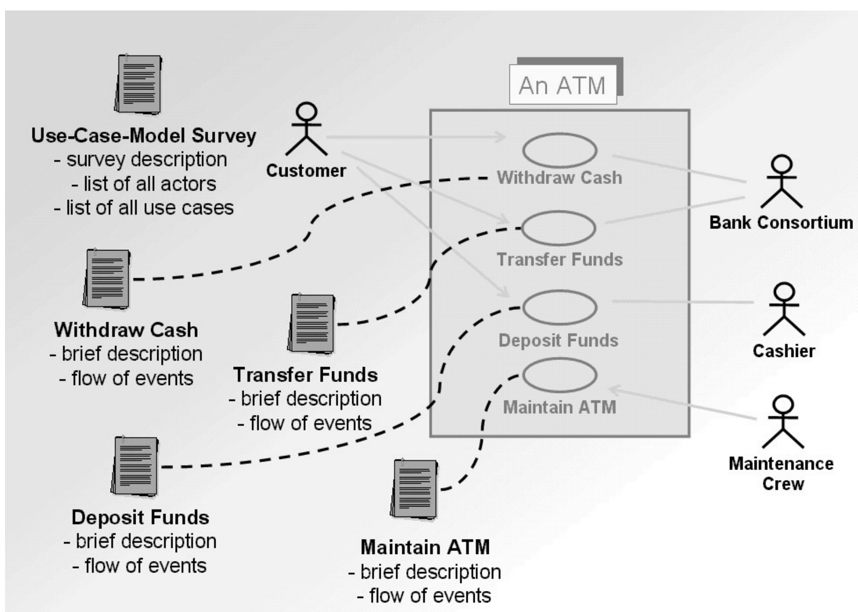


Figure 4.7 Use cases of ATM system.
Reference: (Kruchten, 2000)

Figure 4.7 explains the actors of an ATM system, then the processes and operations within the system (deposit funds, transfer funds, withdraw cash, etc.) for the user or the customer of the ATM machine. The actors like maintenance and setting of the machine are also drawn in the system.

Many development processes that use UML advocate that the system development should start with use case modelling to define the functional requirements on the system. The objective of use case modelling is to identify and describe all the use cases that the actors require from the system. The use case descriptions are then used to analyze and design a robust system architecture that realizes the use cases (this is what is referred to as "use case driven" development). But how the engineers can know that all of the use cases, or even the correct use cases that best support the business in which the system operates, are identified? To answer such questions analysts need to model and understand the system's surroundings. Modelling a business's surroundings involves answering such questions as (Eriksson & Penker, 2000):

- How do the different actors interact?
- What activities are parts of their work?
- What are the ultimate goals of their work?
- What other people, systems, or resources are involved that do not show up as actors to this specific system?
- What rules govern their activities and structures?
- Are there ways that actors could perform more efficiently?

UML and its components are used in common by software engineers but some difficulties and limitations of UML are also declared from their experiences. The UML focuses specifically on the software more than derivation of user’s needs (Pressman, 1997). UML assumes that there exist a document of requirements at the beginning of the requirements modelling and Booch, Rumbaugh, & Jacobson (1998) and (Lopes & Barreto, 2002) considers this assumption as an unrealistic situation. Another difficulty is about the notations that are used in UML. Using the same notation for both specifications and design in the software process makes it certainly more difficult to keep the difference between the real conditions and design (Jackson, 1995). It is with non-functional requirements that this difference is clearer. Non-functional requirements deal with constraints that are imposed on the system itself, or on its development process (Kotonya & Sommerville, 1996). In a good requirements specification they have to be expressed objectively, so that the system can be verified. Designers achieve that by gathering application domain properties that give us a measurable quantity and may not have a direct correspondence in the machine. It is not easy to represent non-functional requirements using the UML notation. This notation cannot be overlooked since non-functional requirements play an equal or sometimes a more important role than functional requirements. Non-functional requirements can easily be modelled using the requirements models of enterprise reference architectures (Jain, Mohan, & Dholakia, 2004).

Lopes & Barreto (2002) indicates some deficiencies of UML Use case diagrams. As stated in the UML specification, use case diagrams describe a sequence of actor stimuli and system responses that are initiated by an actor. Thus, it is not possible to represent interaction situations where the communication is initiated by the system. Another drawback is that it is forbidden to represent actor associations. Without being able to represent such relations, designers miss the part of the real world. Use case diagrams also have the lack of a straightforward way to represent use cases structural hierarchy or the structure between use cases, the lack of adequate means for dealing with use case interaction, the impossibility of automatically maintaining the relationships among the diagram elements, making it very expensive to implement traceability

UML is meant to describe the machine. The lack of a process or guidelines on how to use it as a requirements modelling tool is a huge problem, Other languages, traditionally used in the design phase were successfully used in requirements modelling, e.g., SADT (Vernadat, 1996), which has evolved into the IDEF family of process modelling languages. Others, based on phenomena relevant to the requirements engineering activity, such as goals and agents, are being developed

(Dardene, Lamsweerd, & Fickas, 1993). However, the support for it is still academic and its concepts are not as widespread as the ones of UML.

4.8 User Centered Software Development Techniques

User centered design and development techniques have an increasing trend in software engineering field of study. Different approaches are developed to ensure that user requirements are placed within the characteristics of design issues. Participatory Design (PD), Rapid Application Development (RAD), Joint Application Development (JAD), Joint Requirements Planning (JRP), and Dynamic Systems Development Method (DSDM) are the most commonly used techniques among the user centered design and development approaches.

The main techniques used in these phases are known as Joint Requirements Planning (JRP) and Joint Application Design (JAD). Both these techniques make a heavy use of meetings which the developers and the prospective users work together. JRP and JAD have much in common with a design method known as Participatory Design (PD). Both emphasize end-user involvement. They have differences in defining their goals. User involvement in JRP and JAD is primarily intended to speed up the process of producing the right system (Vliet, 2007). JAD, as the facilitated group process, is established as an alternative to the conventional interviewing technique for determining systems requirements (Dennis, Hayes, & Daniels, 1999). A successful JAD process may reduce the communication barriers to effective requirements elicitation and analysis and eventually help to improve the quality of the final system (Carmel, Whitaker, & George, 1993). However, JAD groups have experienced problems usually associated with the freely interacting meeting structure (Davidson, 1993).

JAD is superior to the interviewing technique and takes significantly less time. During a JAD session a trained facilitator helps system developers and users pool their knowledge to establish system requirements. The facilitator stimulates effective user–developer interactions to generate ideas about system features and helps to speed up decision making (Crawford, 1994). JAD and its many derivatives have become increasingly popular in systems development and other organizational, decision-making contexts. It is considered a best practice for fostering a user commitment. JAD, however, relies heavily on facilitation to guide the meeting toward the attainment of its objectives. Effective JAD facilitation is essential to encourage group productivity, resolve conflict, and minimize problems that are usually associated with freely interacting groups (Duggana & Thachenkary, 2004).

RAD tends to focus on the practical acceptability or the utility of information systems as a key measure of success. The objective is to deliver measurable improvements in organizational performance. In contrast, PD tends to focus on the social acceptability and the usability of information systems. The objective is particularly to increase levels of job content and worker satisfaction. One of the key principles of PD is that developers must concern themselves not only with the development of technical systems but also with the design of work (Davies & Holmes, 1998). JAD is often used with RAD, an iterative and incremental approach for accelerating information services delivery, and with DSDM, a RAD-based technique considering a risk-reducing investment in information systems development (Kumar, 2001).

DSDM defines itself as a user-centered development approach. Active involvement by the user community throughout the development project is therefore seen as crucial to successful development work. In DSDM development teams, both users and developers must be given the power to make key decisions. The developers need to be able to decide rapidly on technical solutions. The business users need to be able to decide upon key requirements for the application. The key emphasis in DSDM is on evolving a system by incremental steps. Partial solutions may be delivered to fulfil an immediate business need (Stapleton, 1997).

4.9 Verification and Validation in Requirements Models

Verification is the process of ensuring that the set of requirements is complete, correct, consistent, clear and reachable. Verification seeks to establish that the subsequent products of the development process accurately reflect the requirements as documented. It is no use taking great care with the requirements only to be able to check that they are carried forward through development, e.g. to the formulation of a testing programme, in a consistent fashion. Software development orthodoxy sets down that at each stage in software development designers should be able to prove that the specification is secure with respect to the preceding specification. There is a great deal of research aiming to establish the means to achieve this (IWWSD, 1993). Clearly, automated support for formal reasoning and proof requires significant further research. Verification becomes a matter of consistency management in which inconsistency is tolerated at certain points in development while at others consistency is checked and enforced (Finkelstein, 1994).

Validation is the process of checking that requirements meet the expectations of the users. Assuming that the acquisition and modelling processes are imperfect, some validation of the products of the requirements engineering process is necessary. Therefore, they must be analyzed in order to establish the extent to which they accurately embody the requirements. Where there is a mismatch between the conception of the stakeholders and the requirements as documented this must be ironed out. The bulk of the work on validation has concentrated on exploration and inspection which are discussed separately below. Much of the remainder has concentrated on providing modelling schemes which are, in some sense, easy to validate (graphical languages and so on). Other work relevant to validation includes the use of scenarios and specification animation. Work on tools which allow multiple views and browsing of complex document and model structures are also significant. Alternative directions are suggested by work on specification critiquing and on annotation schemes for marking errors in specifications (Finkelstein, 1994).

Ideally validation should be as tightly tied to and interleaved with requirements production as possible. However, organizational factors can intervene to prevent this. In such cases the validator may be faced with large amounts of information and no guidance on how to proceed or what questions to ask. Research on methods for providing such guidance and on developing interesting or relevant questions to ask of the products of requirements engineering would be valuable. There are ways to economize in the application of verification. The systems approach entails the following principal steps (Grady, 2007, p.17):

1) Understand the customer's needs.

2) Expand the need into a critical mass of information necessary to trigger a more detailed analysis of a system that will satisfy that need.

3) Further decompose the need, which represents a complex problem, within the context of an evolving system concept, into a series of related smaller problems, each of which can be described in terms of a set of requirements that must be satisfied by solutions to the smaller problems.

4) Prior to the start of detailed design work it is sometimes necessary or desirable to improve team member confidence in their understanding of the requirements or to prove that it is physically possible to produce a design that is compliant.

5) Apply the creative genius of design engineers and the market knowledge of procurement experts within the context of a supporting cast of specialized engineers and analysts to develop alternative solutions to the requirements for lower-level problems.

6) Integration, testing, and analysis activities are applied to designs, special test articles, preproduction articles, and initial production articles that prove to what extent that the designs actually do satisfy the requirements.

CHAPTER FIVE
QUALITY FUNCTION DEPLOYMENT (QFD) and RELATED TOOLS

5.1 QFD Concept

QFD is an engineering method for converting customer wants into quality characteristics for developing product/service design by systematically deploying the relationships of customer requirements into product/service characteristics. It is not just a tool but also a planning process helping a company to focus on customer requirements and underlying needs (Guinta & Praizler, 1993, p.3).

QFD is also defined by group of detailed schemes that convert the customer's quality perception into product characteristics and then convert the product characteristics into production and assembly requirements. In this manner, the customer voice is deployed to all company (Garwin, 1998).

Another definition is that QFD is a tool in which the demands of the customers or market are converted into appropriate technical requirements and actions; and finally to all phases of product (Fortuna, 1988).

The knowledge of customer needs is a "must" requirement in order for a company to maintain and increase its position in the market. Correct market predictions are of little value if the requirements cannot be incorporated into the design at the right time. The team should take the time required to understand customer wants and to plan the project more thoughtfully. The intent of QFD is to incorporate the "voice of the customer" into all phases of the product development cycle, through production and into the marketplace. With QFD, quality is defined by the customer (Akao & Mizuno, 1994). Customers want products, processes, and services that throughout their lives meet customers' needs and expectations at a cost that represents value. The results of being customer-driven are total quality excellence, greater customer satisfaction, increased market share, and potential growth. QFD uses many techniques in an attempt to minimize and make the task easy to handle large numbers of functional requirements that might be encountered. Applications in the range of 130 (engineering functions) and 100 (customer features) were recorded. This bundling of customer's features is a critical step. It requires a cross-functional team that has multiple capabilities such as the ability to brainstorm, evaluate, and revolutionize existing ideas in pursuit of identifying logical (not necessarily optimum) groupings and hence, minimizing the overall list of needs into manageable classes (Yang & El-Haik, 2009).

The thing that makes QFD unique is that the primary focus is the customer requirements. The process is driven by what the customer wants, not by innovations in technology. Consequently, more effort is involved getting the information necessary for determining what the customer truly wants. Once a product is defined, QFD enables the design phase to focus on the key customer requirements, those elements that are defined as being very important to the customer. By addressing these elements, the design phase is shortened to focus on items that the customer really wants. By concentrating efforts, less time will be spent on redesign and modifications. The savings have been currently estimated as one-third to one-half of the time taken using traditional means. If a new product took eighteen months from concept to market, using QFD could reduce the time to nine to twelve months, with little if any changes to the product once it is in the marketplace. For many companies, this can mean many dollars saved not only in development but also in additional income brought in due to getting out a product that meets the customer's needs faster than before (Bossert, 1991).

Since QFD is a customer-driven process, it creates a strong focus on the customer (Shilito, 1994). QFD exercises tend to look beyond the usual customer feedback and attempt to define the requirements in a set of basic needs, which are compared to all competitive information available. Therefore, all competitors are evaluated equally both from the customer's perspective and from a technical perspective. Once this information is in hand, then, through a Pareto ranking, the requirements are prioritized, and the manager can then effectively place resources where they can do the most good—on the requirements that are meaningful to the customer and that can be acted upon (Bossert, 1991).

5.2 Historical Evolution

5.2.1 Brief History

QFD approach was first conceptualized by Dr.Yoji Akao in 1966, during Japanese automotive industrial transition from product imitation to new product development. The first reported case study was by Bridgestone Tire in Japan in 1966 (QFDI, 2006). The first application and publication was put forward in 1972. The first article was written with the title "Standardization and Quality Control" by Dr.Yoji Akao (Akao, 1988).

At the end of 1960s, Japan became a successful country on the manufacturing of steel products with the lowest costs. Japanese manufacturers decided to use this competitive advantage and their strategic industrialization plans by focusing on ship building industry and they became a leader on manufacturing and construction of super tanker cargo ships at the beginning of 1970s all around the World. The manufacturing and construction of such kind of ships was complex and hard to be managed because of the complexity of its motor, manoeuvre, and balance systems. The size of a super tanker was three times more than a football field. Thus, mass production techniques could not be used for such huge projects. Super tanker ships need a projected and well-planned assembly with respect to the differentiating requirements of customers about handling and transporting technology for the various types of ships. Because of these facts, super tanker manufacturing might become a logistic nightmare (Guinta & Praizler, 1993).

Super tankers were manufactured and constructed in the Kobe Shipyard of Mitsubishi Heavy Industries. Mitsubishi requested a governmental technical support to improve the logistic activities of this complex product. According to this request government authorities applied to the related university professors and demand them to develop a system about planning all the manufacturing activities to meet the special customer requirements during the design and manufacturing processes. Among the selected professors, Dr. Yoji Akao and his project colleagues developed a systematic approach to convert the special customer requirements into design and manufacturing processes. This approach was then defined as “Hin shitsu Ki no Ten kai” in Japan and “Quality Function Deployment” in English. Figure 5.1 is the corresponding Japanese words standings for QFD (Yenginol, 2002).

品質機能展開

Hin shitsu Ki no Ten kai

Figure 5.1 Japanese description of QFD.
Reference: (Yenginol, 2002)

Dr.Yoji Akao, who managed successful QFD applications played an important role on the improvement of total quality management (TQM). Furthermore, he gained the Deming Prize in the later with his QFD studies in TQM. TQM thought extremely affected the Japanese Production Style starting from 1950s and spread all over the country till 1960s. At the end of 1960s, the concept “customer” became an important issue in terms of designing studies concerning the quality perception. Ishikawa/fish bone diagrams were used to find out the customer needs. However, as the products were getting more complex, these diagrams got much larger than they could manage and could not provide adequate and efficient information to designers. In 1966, Dr. Akao declared that the critical points including customer needs should be defined to assure the quality in design and manufacturing. Therefore, this declaration formed the main infrastructure and the basic idea of QFD methodology. Then the first implementation was carried out in the design and manufacturing processing of super tankers in 1972. In those years the major component of QFD was the quality matrix. Dr.Akao, Dr.Mizuno, and Dr.Furukawa first applied the quality matrix in this project (Shilito, 1994).

In 1975, QFD research group was established and three years later, first QFD book was published in 1978 namely "Quality Function Deployment" by Mizuno, Shigeru, and Akao. QFD was introduced to Toyota in 1979 and applied to Hino Motors and Auto Body Group in the same year. The first symposium was organized in Japan, by Japan Productivity Center (Akao & Mizuno, 1994).

QFD spread to North America at the beginning of 1980s when the basic thoughts of QFD began to be talked about. Glenn Mazur translated QFD books into English and the U.S. met with the QFD concept. Richard Zultner, who concentrates his studies on the application of QFD in software developments, took the leadership about QFD applications with Glenn Mazur (QFDI, 2002).

The first US application "Company-Wide Quality Control and Quality Deployment" was sponsored by Cambridge Corporation and co-sponsored by American Society for Quality Control and started applications besides developing the method. It was 1984 when the first application was carried out in the USA. During these years, Japanese applications were expanded to different design areas, e.g., software development. In 1987, the first presentation was carried out in Europe by Galgano & Associates, Italy. At the same time, Japanese Standard Association published QFD Case Studies Book. QFD was first introduced in Brazil, in International Conference on Quality Control, Rio de Janeiro, 1989. Then, QFD began to be distributed all over the world by spreading out its benefits in real applications and academic researches (Akao & Mazur, 2003).

In 1990s, QFD Institutes and International Council for QFD (ICQFD) were started to be established. Today, there are seven QFD Institutes (US, Japan, Brazil, Mexico, Germany, Australia, Sweden). These institutes are the authorities in their region, about consultancy and training support (Mazur, 1991).

The major benefit of QFD is customer satisfaction. QFD gives customers what they want, such as shorter development cycles, avoidance of failures and redesign peaks during prelaunch, and "know-how" knowledge as it relates to customer demand that is reserved and transferred to the next design teams (Yang & El-Haik, 2009).

According to Hauser & Clausing (1988), QFD was originally developed to solve three problems generally diffused in Western industry: (1) the customer's voice was held to be of no account; (2) a considerable loss of information occurred during the cycle of product development; and (3) the different interpretations were given to technical specifications by the various departments involved. Furthermore, QFD supplies the solution to two problems closely related to those mentioned earlier: the *subdivision into departments* and the *temporal serialization of activities.* The application of QFD on a horizontal plane within the organization reduces the negative effects of departmental subdivisions. The members of a QFD team work *together* and not as separate entities. One of the most known benefits of QFD is its ability to *generate and maintain involvement within the work team* over the whole product development cycle. The results of the ensuing synergy are greater than the sum of those obtained by single components. Pooling knowledge within the work team leads to improved decisional capabilities and favours the disappearance of personal prejudices. The short-term benefits brought by QFD include shorter product development cycles, fewer modifications in planning, fewer initial problems, and improved quality and reliability (Francheschini, 2002).

Many companies, especially in Japan and in the United States, have benefited from QFD in that it has been instrumental in achieving notable improvements in planning cycles while at the same time attaining reduced product development times and costs. For example, Toyota Auto Body Co., Ltd., in Kariya, Japan, witnessed an overall reduction of 61% in the initial costs involved in introducing four new models of vans between January 1977 and April 1984 (Hauser & Clausing, 1988). Furthermore, QFD contributes to the creation of a solid platform of *Basic knowledge* in planning. Once the method has been successfully applied in a project, the platform of basic knowledge thus created becomes a data bank storing technical information of extreme importance. The tables and documents prepared during a QFD process constitute a work documentation that becomes a source of ready reference, from which to glean new and interesting ideas for future projects (Francheschini, 2002).

New studies are maintained and QFD is applied to the new industrial design projects as new trends are arisen in different countries (Mazur, 2008). Turkey has not established a QFD Institute yet, but various studies have been handled in both industrial and academic environment. Dokuz Eylul University (DEU) Faculty of Business is a member of ICQFD and attends the annual meetings as a representative on behalf of Turkish QFD practitioners and academicians. DEU Faculty of Business organized two national and an international QFD symposiums with the contributions of Professors of other universities who applies QFD and other quality tools. The further studies to spread the methodology all over the country have been sustained with the help of distinguished academicians in DEU and other Universities in Turkey.

5.2.2 Traditional QFD

In the first applications, a structured table was used, called "quality matrix", which includes the relationships between quality perception of customers and design characteristics. QFD and quality matrix have been improved since their introduction. QFD became a stepwise systematic approach with a detailed preparation phases, and the quality matrix was expanded with respect to the required additional comparisons and analysis (Day, 1997). This stage was defined as the second generation of QFD. This generation involves the following steps and analyses:

1) Define Goals of the company or the product (fishbone diagrams can be used to define the goals).
2) Define Customers In 5W1H table including demographic characteristics.
3) Define target customer segment(s).
4) Collect and analyze the voice of customers (Gemba analysis, interviews, ServQual questionnaire, Kano model, benefits-features table).
5) Retrieve customer needs from observations and verbatim of customers.

6) Organize the needs into groups (Affinity diagram).
7) Construct the hierarchy of needs and find the importance of them (Analytic Hierarchy Process).
8) Handle the weighted needs to the House of Quality matrix.
    a. Define the technical characteristics (columns),
    b. Compare the current product and competitor's product with respect to the customer needs, set the sales points and target level (benchmarking and planning part)
    c. Evaluate the relationships between customer needs and technical characteristics (relationships)
    d. Determine the relationship in each couple of technical characteristics (Correlation part, the roof)
    e. Calculate the final weights of technical characteristics (See section 5.3.2).
    f. Define the design target considering the weights of technical characteristics and contradictions between these characteristics if occurred.

The QFD matrix became an integrated table including benchmarking, planning, relationships, technical characteristics, and design targets . Contradictions between the technical characteristics are also considered in this table, which constructs a roof at the top of the table. This appearance is similar to a house, thus this integrated table is called "House of Quality (HoQ)".

Figure 5.2 is a standard HoQ including seven parts. In this section only brief information is introduced about House of Quality, then in section 5.2.3 detailed information is presented including details about calculations.

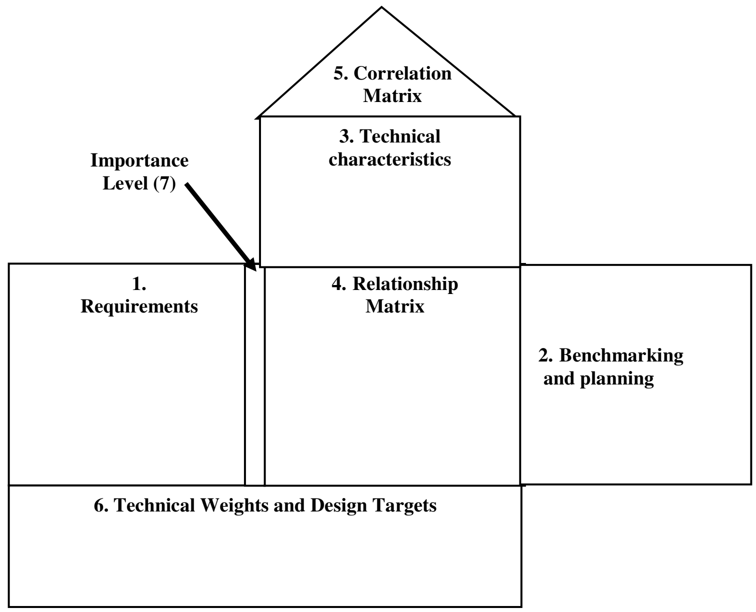


Figure 5.2 The house of quality matrix of QFD.
Reference: (QFDI, 2002)

In the matrix in Figure 5.2, first the customer requirements are determined and classified according to the affinities with each other. Then weights of these requirements are calculated to define the importance level. In the next step, technical characteristics are listed, and if needed, they are classified and then using a scale or another quantitative method, the relationships between requirements and technical characteristics are determined. Benchmarking and plan matrix part consists of comparisons with the competitors due to the requirements and here, planning values are calculated by using a scale. Ordinal scales are used to evaluate different parts of the matrix, e.g. for relationship matrix four-level ordinal scale is used as 0: no relationship, 1: weak relationship, 3: strong relationship, and 9: very strong relationship. After the evaluations are translated by the means of numbers, the matrix values are manipulated in order to reach technical weights and targets by using weighted sums and normalizations. The correlation matrix which constructs the roof of the house shows the relationships among the technical characteristic to see the contradictions if exist. That is, if a technical characteristic is negative correlated with another one, and then improving one of them would affect negative impact on the other characteristic. At the end, the target values for technical characteristics help decision makers to determine which technical characteristics should be improved.

QFD evolved during the years, and new matrices have been added with respect to the number of continued deployment processes, i.e., product planning, process planning, operational requirements, technology, reliability, suppliers, cost analysis, etc. The most commonly known presentation of multi-phase matrix application was the four-phase model. In QFD training activities the four-phase model is presented as an example, but in practice not that many companies have used it exactly the same. Figure 5.3 is the structure of the four-phase QFD model including four serial matrix applications. Each matrix uses the outputs of the previous matrix as the inputs of the current phase.

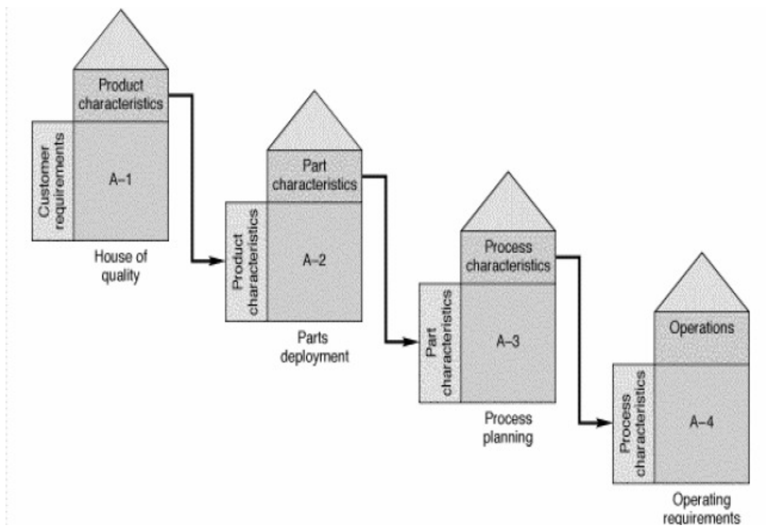


Figure 5.3 Four-phase model.
Reference: (Russell & Taylor, 2003, p.99)

The third generation of QFD, explained in section 5.3.2, implements the basic thought of QFD as in the previous generations. Nearly, all steps followed in traditional and Modern QFD methodologies are performed for the same purpose, but the way and tools used in these steps are modified in Modern QFD. Besides, mathematical corrections are made on the scale of evaluations. In traditional QFD, the main focus is in matrix constructions and calculations, but in Modern QFD the main focus is on the preceding steps. These matrices are used in complex design processes. New tables are added before matrices, and design targets can be retrieved from these tables if the product is not so complex. To avoid from repetitions, the detailed steps are explained in Modern QFD part (section 5.3.2).

The QFD methodology integrates different techniques to meet the expected functionality during the conversion of customer needs into design characteristics. Among the analytical techniques, Analytic Hierarchy Process, Failure Mode and Effects Analysis, Pugh Concept Selection, Theory of Constraints, Taguchi Design, TRIZ, New Lancaster Strategy, Concurrent Engineering, and related TQM tools, i.e., the seven management tools are most commonly used within QFD projects to cope with the difficulties and contradictions (Terninko, 1997).

5.2.3 Modern QFD

In recent years, traditional QFD has changed a little upon its applying methodologies. The concept is the same, but in the methodology, there exist some contradictions and cumbersome structure. Also, Six Sigma applications need QFD and it should be integrated with it. However, the mathematics of the traditional QFD cannot be integrated into six sigma phases. As a result, QFD is modernized and called "Modern QFD", based on Blitz QFD, which also uses house of quality but through more tailored process changing according to the organization. As its name brings to mind, the improvements concentrate on the fast and effective implementations. In the traditional approach similar phases were carried out but, Yet, Modern QFD approach makes these phases more practical and rapidly usable. Summary about the difference between traditional QFD and Modern QFD is explained in Table 5.1.

Table 5.1 Modern QFD for 21st century & six sigma

Traditional QFD	Modern QFD
Canned QFD Process -House of Quality only -4 phase model	Tailored QFD process -based on the voice of the company
Math that works in Japan (only)	Math that is correct and valid
Few big Labourious tools (mostly matrices)	Many small focused tools, matrices rarely
Focused on comprehensiveness	Focused on speed and efficency
Lip service to Voice of customer concept	Voice of customer is the central concept

Reference: (QFDI, 2006)

In early 1980s, the QFD Institute began a special research Project: to apply QFD to QFD. After investigating over many years, North American firms were doing with QFD –and not doing with QFD, a number of significant revisions were made to the basic approach to QFD. This led to the development of the Modern QFD® approach (QFDI, 2006)- a freshen streamlined way to get the basic benefits of QFD with minimum effort. New tables and techniques are adapted and design tables are developed to make the conversion process easier. Figure 5.4 presents the framework and steps of Modern QFD.

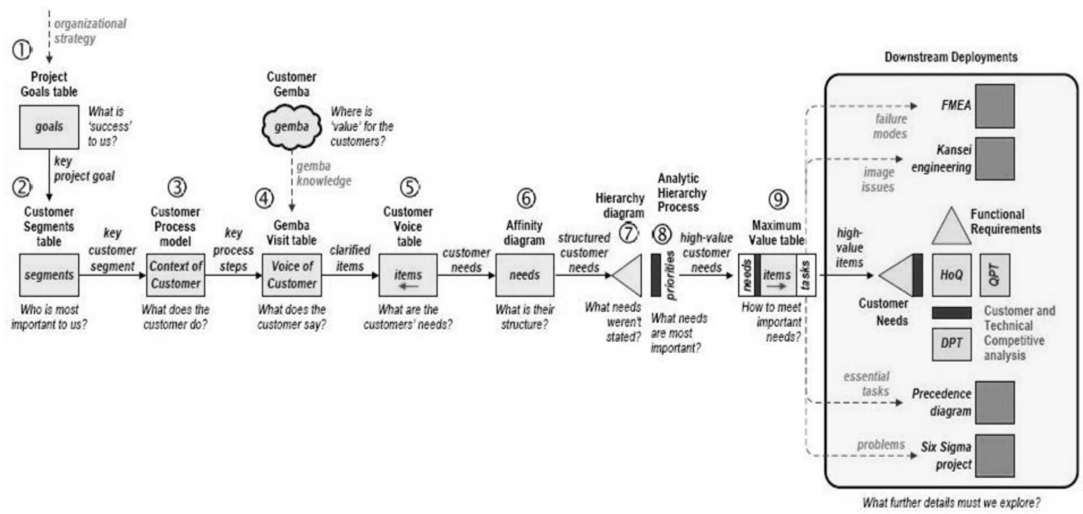


Figure 5.4 Modern –QFD approach.
Reference: (Jayaswal, Patton, & Zultner, 2007, p.28)

Another problem with the QFD applications was about the scale that is used in the matrices. Traditional QFD matrices multiply ordinal scale with ratio scale and normalize it, which is mathematically invalid. Because of that problem, QFD cannot be well integrated with Six Sigma. Then, for the relationships, instead of using ordinal scale, ratio scale generating mechanism was developed using the analytical hierarchy process (AHP) method. This scale is validated by Metrology professors in US (QFDI, 2006).

Section 5.3 presents the application steps of Modern QFD in details in accordance with the examples from the case studies.

5.3 Modern QFD Steps

5.3.1 Initial Steps to analyze Voice of Customer

QFD evolved into an integrated methodology and reached its third generation which is called "Modern QFD based on Blitz QFD®" (QFDI, 2006). Some new tools have been appended to its basic development process. In traditional QFD, the ordinal scale is used, which creates some risks about the acceptance by the scientist. In the Modern or third generation QFD, one type of ratio scale is preferred to perform mathematical and statistical operations.

Through the Modern QFD, requirements analysis is performed with a few modifications in the preparation phase (steps 1 to 7 in the following). In the last phase, the content and the name of the QFD matrices are developed from scratch for enterprise modelling aspects in Enterprise-QFD, the proposed methodology for requirements engineering. The blitz QFD-based calculations are then performed. The QFD calculations are improved to avoid its sceptical mathematical scale, which is changed from ordinal scale to ratio scale. Blitz QFD presents only the tools for analysis especially for product and service design. Blitz QFD first analyzes the users' or customers' verbatim and clarifies the requirements and needs. Then using evaluation matrices, converts these needs into design characteristics.

Blitz QFD handles each requirements analysis study as a project, and starts the steps by defining the goals of the project. In the second step, user segments are determined with the characteristics and the management decides which segment(s) will be considered in the modelling. In the third step, the customers' verbatim is collected from the target customer segment(s). These steps are followed by clarification of the verbatim; analysis of the verbatim with respect to the customer needs; the prioritization of the needs, and the transformation of the needs into modelling issues through the evaluations, specified tables and further sequential matrix calculations. The details about how these steps are presented are as follows (QFDI, 2006; Jayaswal, Patton, & Zultner, 2007):

1) Project Goals:

Product design processes are handled as a project management perspective in which goals of improvement on product or even the goals to design new product are defined and traced. The goals are declared by answering the questions “how is the goal measured?”, “in which time frame?”, “who will judge the result?” Thus, the goals are traced and managed with measurable scales and predefined responsibilities. Table 5.2 is an example of a project goals table for a design process in education industry where the business education of a university is reviewed for active learning concept. Table 5.3 is compiled from another application about process design in shipping line industry where the operation process is analyzed and improved.

Table 5.2 Project goals for business education review project

	Goal Statements	**How measured?**	**Time Frame**	**Who judges success?**
Final Project Goals	to be more preferable Faculty in University entrance exam	University entrance exam rankings	2 years	**Council of Higher Education in Turkey**
	to enlarge the rate of job placement of graduates of Department of BA	Job placement statistics of Faculty	4 years	Faculty Administration
	to increase the satisfaction levels of students	Survey regarding students’ satisfaction	1 year	Faculty Administration
	to increase the satisfaction levels of academic staffs	Survey regarding academic staffs’ satisfaction	1 year	Faculty Administration
	to increase the satisfaction levels of companies	Survey regarding companies’ satisfaction	1 year	Faculty Administration
Project Goal for this study	to improve “skills” and “attitudes/behaviours” of students of Department of Business Administration	Survey regarding skill and attitude/behaviour improvements (It will be applied in every year to follow changes in improvement)	4 years	Faculty Administration

Reference: (Ozgen, Kurt, & Ozdagoglu, 2006)

Table 5.3 The project goals for shipping line "operation" process

Project	Analyzing and improving the operation process			
	Goal statement	How measured?	Time frame	Who judges success?
Project goal	Increasing the satisfaction level of charterers or brokers	Measurement of customer satisfaction by a questionnaire The number of ships chartered	One year	Chartering Manager and Operations Manager

Reference: (Kapucugil Ikiz & Ozdagoglu, 2008)

2) Identify User Segments:

Same product or service may be related to different customers. Therefore, characteristics of the product differ with respect to each customer segment and segmentation should be carried out before the design phase to differentiate the product according to the needs of each segment. Furthermore, constraints and capacity limits may force the designers to choose only one customer segment. For all these situations, QFD specifies its customers with segments via a structured table called "customer segment table". Two examples are presented here from the different types of industrial implementations. Table 5.4 is an example of a customer segment table for a design process in education industry where the business education of a university is reviewed for active learning concept. Table 5.5 is from another application about process design in shipping line industry where the operation process is analyzed and improved.

Table 5.4 Customer segments table of business education review

	Who uses project?	What is project used for?	When is project used?	Where is project used?	Why is project used?	How is project used?
Internal Customers	Students	To learn	When they select Department of BA, Faculty of Business, DEU	In education process	To improve learning capacity of students, skills and attitudes	
	Academic Staff	To teach	When they meet their students in 2006-2007 academic period.	In education process	To improve learning capacity of students, skills and attitudes	
External Customer	High School students	To select right university, faculty and department	When they plan their education	In university selection period	To reach more interactive education system	
	Private organizations and State organizations	To employ qualified personnel	When they need qualified personnel	At their organizational structure	To operate their business functions (marketing, accounting, finance, human resource, production, purchasing etc.)	By using human resource department and employee selection techniques

Reference: (Ozgen, Kurt, & Ozdagoglu, 2006)

Table 5.5 Characteristics of the customers for shipping line "operation" process

Who uses the operation process?	What is the operation process used for?	When is the operation process used?	Where is the operation process used?	Why is the operation process used?	How is the operation process used?
Charterers (external customer)	Chartering Negotiations	When they need	At ports	Loading– Unloading (transportation)	Direct contact with ship-owner company or With help of Brokers
Freight Shipping Brokers (external customer)	Chartering Negotiations	When charterers need	At ports	Loading– Unloading (transportation)	Direct contact with ship-owner company
Office Personnel (internal customers)	Keeping things running well on during the voyage	When a vessel has fixed	In company	Providing all requirements of the ship	Direct contact with both operation unit and captain

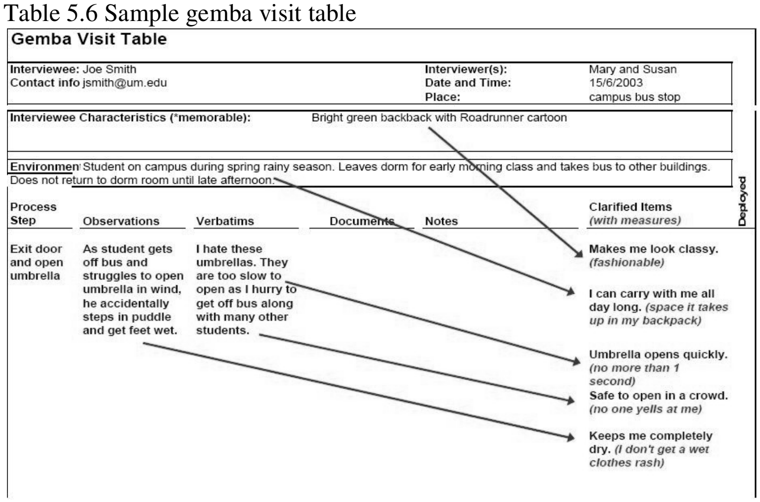
Reference: (Kapucugil Ikiz & Ozdagoglu, 2008)

3) Go to gemba:

Gemba is the place where the product becomes a value from the viewpoint of the customer (QFDI, 2002). Therefore, it is the place where customers use the product, or the place where the product is processed (for internal customers). The gemba, which covers the user segment(s), is observed to discover the user needs. Gemba analysis can be handled via observations, interviews, focus group studies, and questionnaires. The key concept is to "discover" the needs, and the most important phase to analyze gemba is to observe the customers' behaviours, reactions, mimics, and movements while they use the product or get the service. Observations are important in that they enable the analysts to find out the situations or problems which are not declared by the customers. Especially, the questionnaires are indirect resources of gathering the customer voice and have the least priority to choose as a tool.

In Modern QFD the observations and interviews with the customers are arranged in a structured table form called the "gemba visit table" to systematize the records in an observation process. Each column is independent from each other and used to clarify the voice of the user observed. Table 5.6 is a partially selected sample gemba visit table showing a partial interview with the quality system manager of a company about quality items and characteristics.

Table 5.6 Sample gemba visit table



Gemba Visit Table

Interviewee: Joe Smith
Contact info jsmith@um.edu
Interviewer(s): Mary and Susan
Date and Time: 15/6/2003
Place: campus bus stop

Interviewee Characteristics (*memorable): Bright green backback with Roadrunner cartoon

Environmen Student on campus during spring rainy season. Leaves dorm for early morning class and takes bus to other buildings. Does not return to dorm room until late afternoon.

Process Step	Observations	Verbatims	Documents	Notes	Clarified Items *(with measures)*	Deployed
Exit door and open umbrella	As student gets off bus and struggles to open umbrella in wind, he accidentally steps in puddle and get feet wet.	I hate these umbrellas. They are too slow to open as I hurry to get off bus along with many other students.			**Makes me look classy.** *(fashionable)* **I can carry with me all day long.** *(space it takes up in my backpack)* **Umbrella opens quickly.** *(no more than 1 second)* **Safe to open in a crowd.** *(no one yells at me)* **Keeps me completely dry.** *(I don't get a wet clothes rash)*	

Reference: (QFDI, 2006)

4) Customer Process Model:

A Customer process model describes a flow in which the steps, route, and relationships are defined for a particular product or service, and the customer. A customer process model is constructed through data flow diagrams, process flows, and other tools to determine common level of customer understanding. These tools are used to validate the model, i.e., they warn the analysts when they are out of the gemba. Users cannot always articulate what they need, which makes it difficult to perform the gemba visits. In this case, some user stories can be created to discover what the users really mean. All the statements noted are then translated into clarified items to be validated by the users.

5) Customer Voice Table (CVT):

CVT arranges the requirement statements by decomposing them into benefits, needs, and product features gathered directly from the gemba visit table or from complaint reports, warranty data, and sales reports. CVT is also essential for requirements analysis to see which statement means what, and how they are related to each other.

Verbatim of users are converted into clarified items according to the statements gathered from gemba visits. The related columns are then matched to construct some paths through the table. CVT represents the users' verbatim rather than designers' ideas. All possible paths are formed following the related verbatim from left to right and right to left to reach the related need. Figure 5.5 is a partial example for CVT indicating the customers' verbatim for a campus design review study. Figure 5.6 also shows a partial CVT table in a process improvement study of Shipping Line Company. The columns are filled in independently from each other.

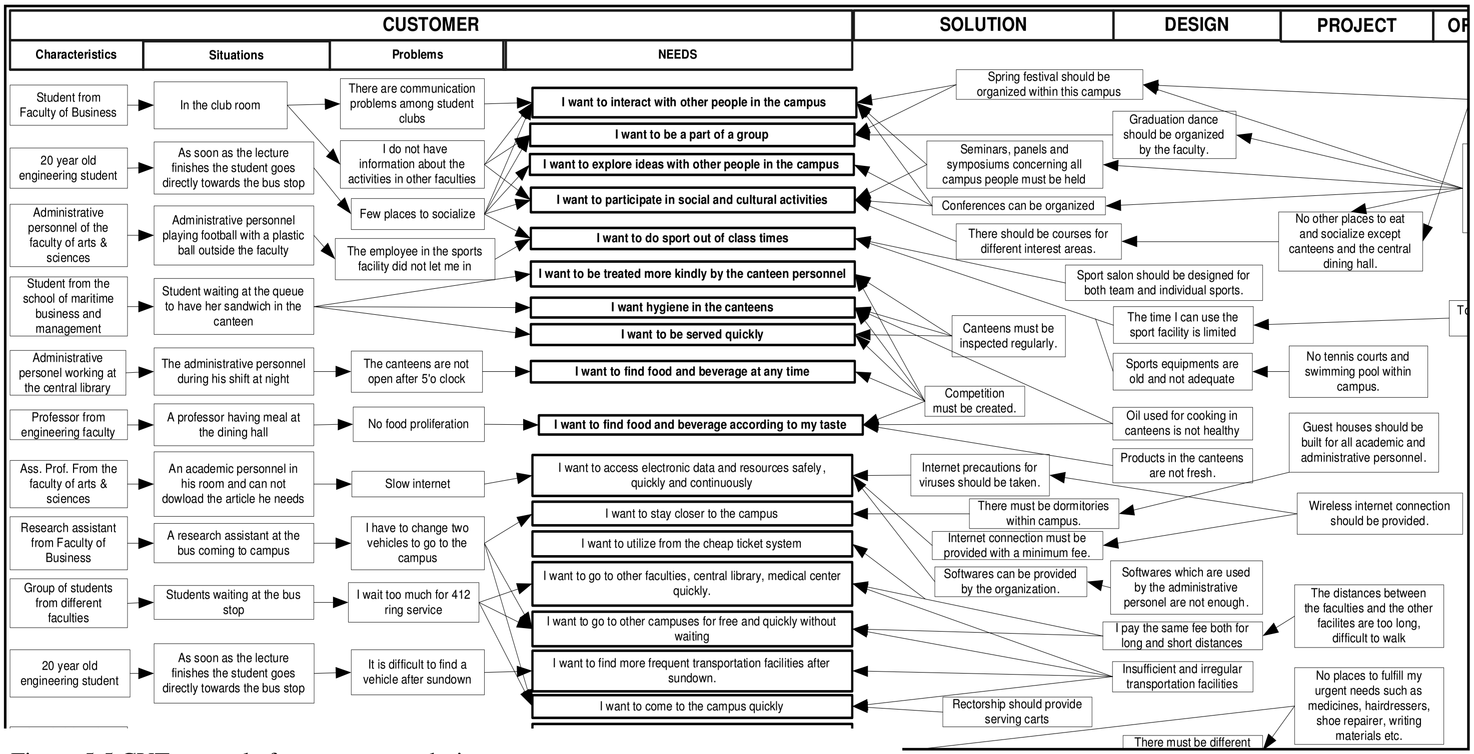


Figure 5.5 CVT example for a campus redesign.
Reference: (Kapucugil, Atrek, Ozdagoglu, & Tüzemen, 2006)

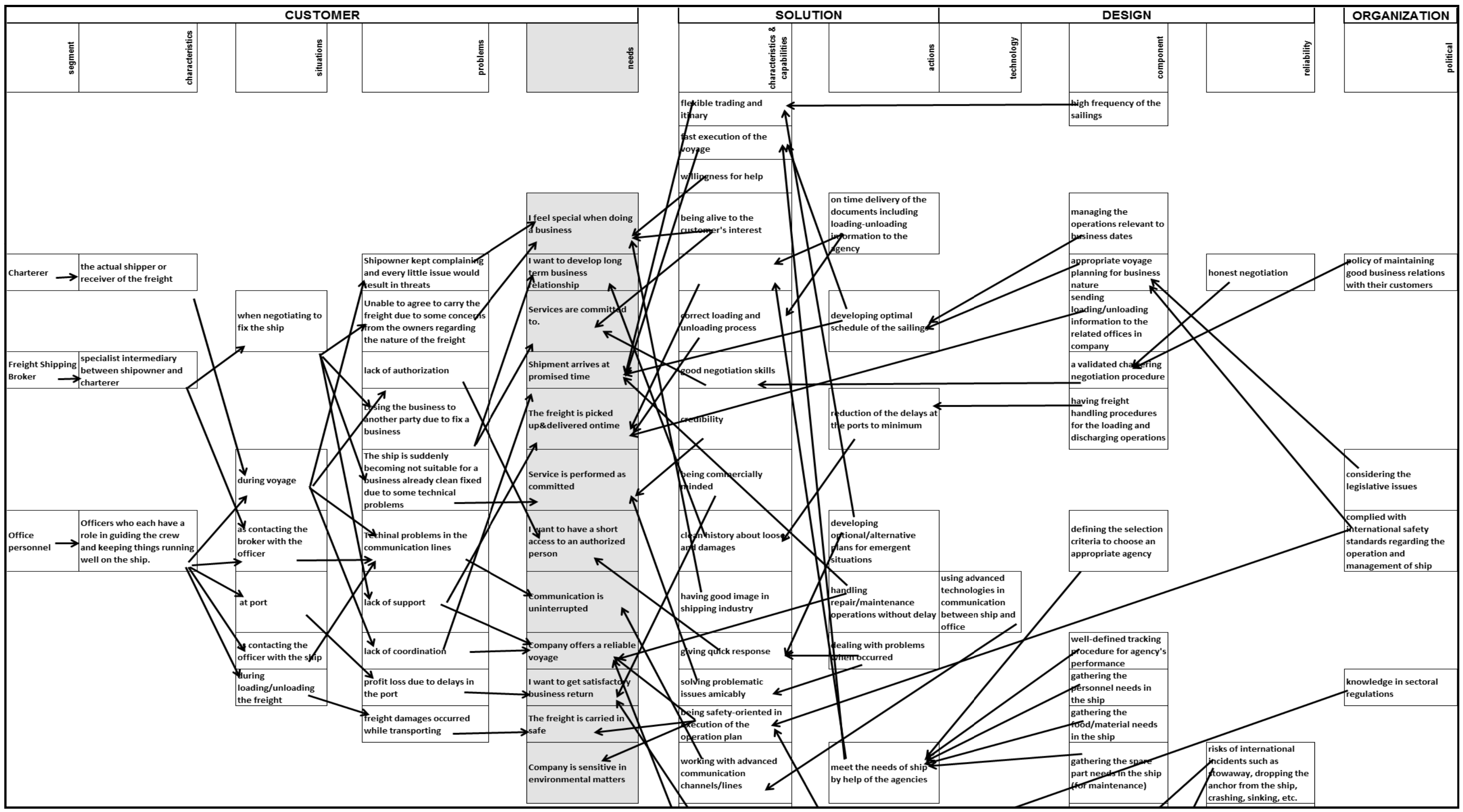


Figure 5.6 Customer voice table.
Reference: (Kapucugil Ikiz & Ozdagoglu, 2008)

6) The structure of the needs (affinity diagram), hierarchy diagram, and priorities:

Analysts collect many various verbal statements indicating many different needs. Because of the constraints and limitations or just for classifying, the needs should be prioritized. These needs are classified before prioritization to prevent from inconsistencies. Affinity and hierarchy diagrams help to classify the clarified requirements in mutually exclusive groups and construct the hierarchy structure within these groups. Figure 5.7 represents an affinity diagram and corresponding hierarchy structure retrieved from a QFD project.

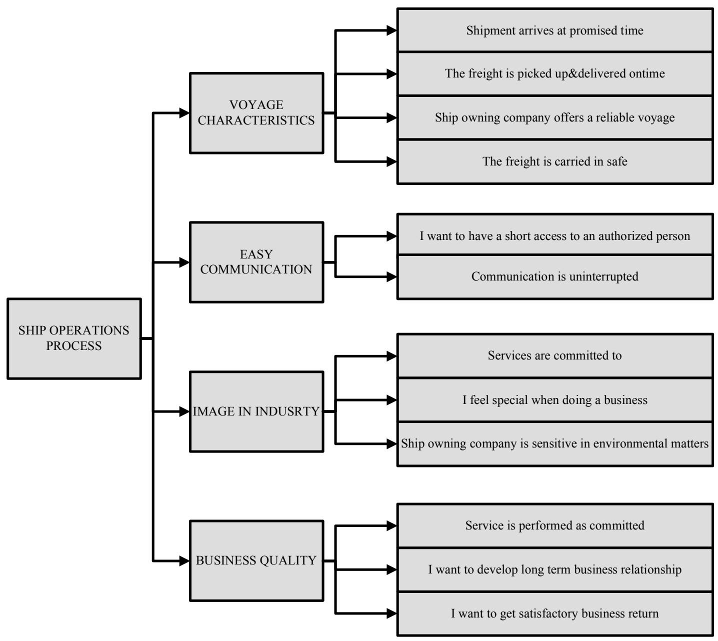


Figure 5.7 The hierarchy of customer needs.
Reference: (Kapucugil Ikiz & Ozdagoglu, 2008)

Because requirements are not equally significant in a hierarchy level, some importance levels should be calculated after the classification through a multi-criteria decision making tool, e.g. AHP, Analytic Network Process (ANP), and their fuzzy extensions which are generally preferred and used by QFD practitioners (Feyzioglu and Buyukozkan, 2008). Construction of a network and calculation of complicated matrix in ANP is so difficult and inefficient that QFD practitioners do not prefer to use this methodology. The prioritization process should not be longer than the requirements analysis and representation for an efficient design process, thus, customer needs are tried to be defined by independent statements and prioritized using AHP in practice. These techniques are explained in section 5.6. Table 5.7 is the corresponding importance table obtained from the AHP technique.

Table 5.7 The hierarchy of customer needs and the prioritization

primary customer needs	global priority	secondary customer needs	local priority	global priority	rank	renormalized priority
VOYAGE CHARACTERISTICS	**0,602**	**Shipment arrives at promised time**	0,228	**0,137**	**2**	**20,0%**
		The freight is picked up&delivered ontime	0,146	**0,088**	**4**	**12,8%**
		Company offers a reliable voyage	0,165	**0,099**	**3**	**14,5%**
		The freight is carried in safe	0,462	**0,278**	**1**	**40,5%**
EASY COMMUNICATION	**0,157**	I want to have a short access to an authorized person	0,468	0,074	7	
		Communication is uninterrupted	0,532	**0,084**	**5**	**12,2%**
IMAGE IN INDUSRTY	0,088	Services are committed to.	0,587	0,052		
		I feel special when doing a business	0,319	0,028		
		Company is sensitive in environmental matters	0,095	0,008		
BUSINESS QUALITY	**0,153**	Service is performed as committed	0,521	0,080	6	
		I want to develop long term business relationship	0,273	0,042	8	
		I want to get satisfactory business return	0,206	0,032	9	

Reference: (Kapucugil Ikiz & Ozdagoglu, 2008)

7) Maximum Value Table (MVT):

This step is applied to share the information gathered from the customer and arranged in the CVT with the model designers. In the maximum value table, key customer needs are driven forward to the various dimensions of design issues by designers. Columns start with the same set used in customer voice table, but new columns may be added to deliver value to the customer. This table provides the designers with the areas that have greater complexity or uncertainty, and where matrices need to be constructed between two design dimensions and the required level of detail for the further analysis

MVT is constructed to classify what functional requirements exist, and what kind of features should be specified (functions, tasks, processes, and entities) from the designer's perspective. It is the road map of designers. MVT was introduced by blitz QFD, and is not available in the classical QFD. Figure 5.8 and Figure 5.9 are samples of MVT representing the design route map for operations process in a shipping company and campus redesign study. This table is modified with respect to the process modelling perspectives similar to Enterprise-QFD approach proposed in section 6.1.

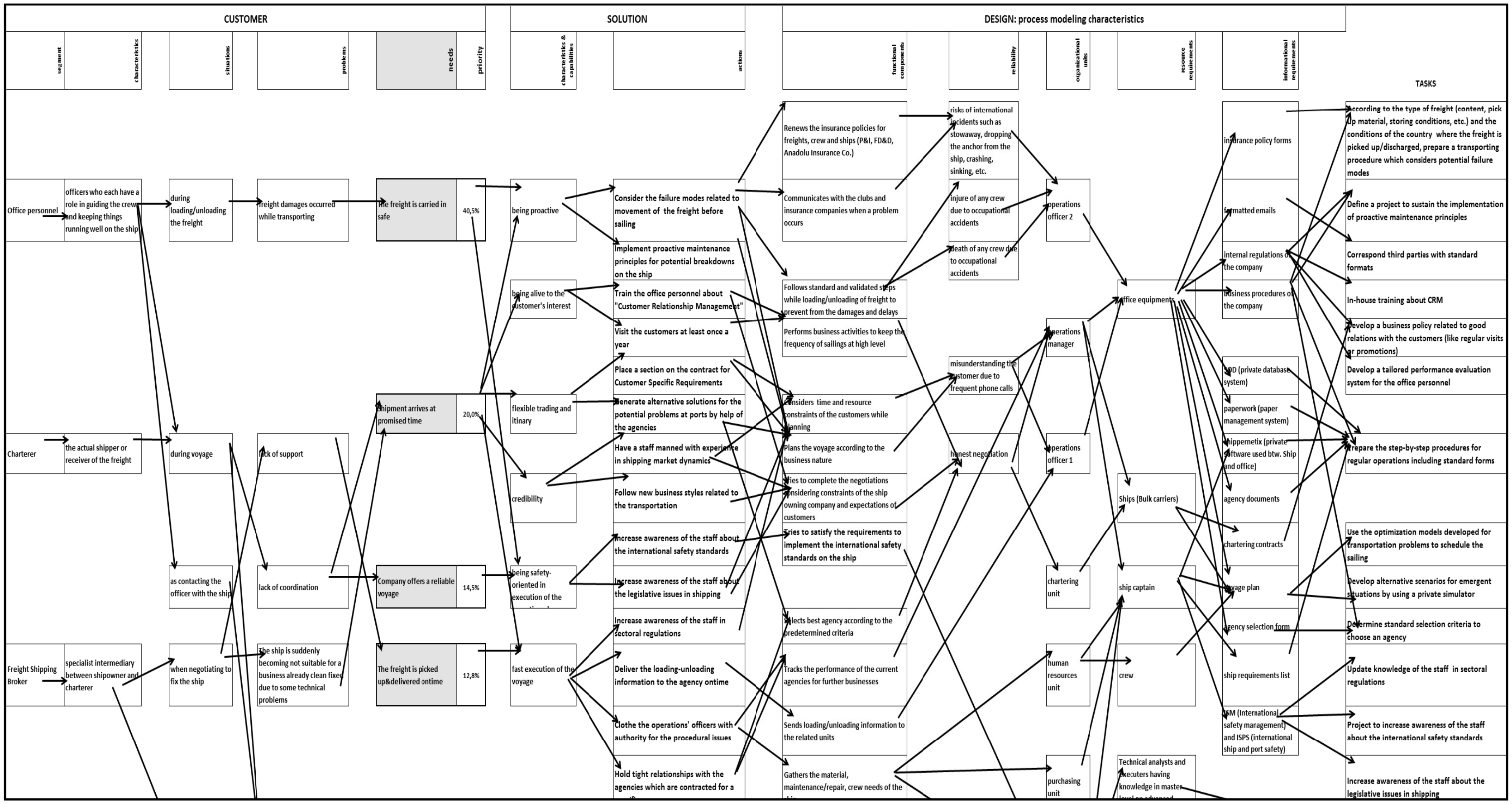


Figure 5.8 Maximum value table example.
Reference: (Kapucugil İkiz & Özdağoğlu, 2008)

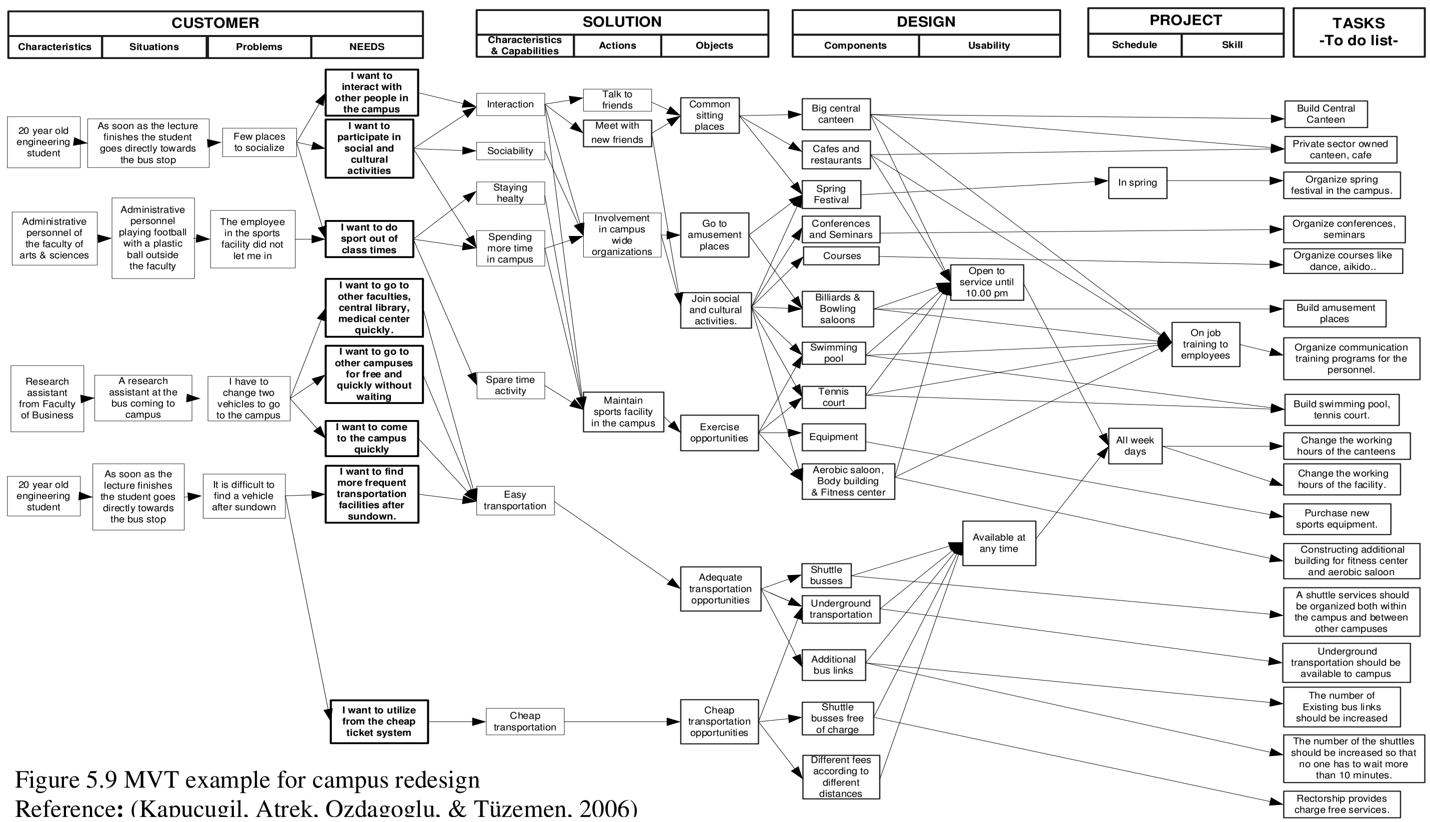


Figure 5.9 MVT example for campus redesign
Reference: (Kapucugil, Atrek, Ozdagoglu, & Tüzemen, 2006)

5.3.2 QFD Matrix Structure and House of Quality

The QFD process is broken down into numerous analytical steps; preparation steps for gathering and discovering the needs, and then analysis steps converting them into design specifications including systematized tables, charts, and matrices. The QFD matrices depend upon some functional relationships through which the developer can quantify quality and establish priorities. After the preparation phase, the conventional QFD constructs these matrices in a four-step process: quality design, detailed design, process deployment, and production planning. As the shape of the matrix looks like a house, it is also called “House of Quality (HoQ)”. Classical QFD collects data from “Customer Voice”. It then converts customer requirements into technical product or service characteristics through the QFD matrix. Figure 5.2 in section 5.2.2 also shows its general structure. This matrix is then followed by other matrices for the requirements analysis process (QFDI, 2002).

QFD employs matrices for detailed requirements analysis that translates quality into design features. After customer requirements are determined, classified and weighted, the planning and benchmarking is carried out to benchmark the product and propose some improvements in the product design. At the end of the planning and benchmarking, the final weights for each requirement are defined. The next phase after benchmarking and planning consists of determining the design specifications of the product and the relationships among these specifications in the roof of the house. The relationship matrix is formed to define the relationships between the customer requirements and design specifications. The importance of each design specification is then assessed through the manipulation of customer requirements weights and relationship values. The bottom of the house represents the operations about the design items. Based on the benchmarking results, and considering the verbal correlations in the roof, design targets are set for the next matrix in the sequence.

The integrated matrix in Figure 5.2 consists of seven sub-matrices, namely, customer requirements list (1), technical characteristics (3), importance levels (7), relationship matrix (4), benchmarking and planning (2), technical weights and design targets (6), and correlation matrix (5). After a detailed preparation phase including project goals, customer segmentation, customer voice analysis and clarifying the needs (see section 5.3.1), the requirements list (part 1) is defined as the first part. Then, the needs are grouped into categories and the hierarchy of these categories is determined. The customers evaluate the needs and assign values using the procedure of Analytical Hierarchy Process (AHP) (Saaty & Vargas, 1994) and related methods (part 7), e.g., Analytical Network Process (ANP) (Saaty, 2001). These weights constitute the importance values part in the matrix. The first two parts are the inputs of this matrix and when they are ready to use, the evaluation phases are carried out. First of all, the measurement scale is constructed for each different evaluation. The verbal statements are defined and then weighted by each customer by using AHP or related methods, and therefore ratio scale numbers are obtained for each type of evaluation (see section 5.5). After the technical characteristics are defined in the columns (part 3), each of them is evaluated by each customer requirement in the rows and the relationship values (part 4) are assigned with respect to the evaluation scale. In the next phase, each requirement is handled to be compared with the competitor(s) and planning values about improvements are defined (part 2) according to the corresponding ratio numbers of verbal statements. If enterprise engineers need additional parts such as technical advantages and challenges, then these parts may be added and further evaluations may be performed about them. The calculations are handled according to the linear distribution of ratio scale values. The final numbers in the matrix present the importance values of each technical characteristic (part 6) with respect to the customer preferences and competitors status in the market. The roof of the matrix (part 5) is used to show whether there is a contradiction between each couple of technical characteristics by the view point of the design process. The QFD practitioner collects the importance values and combines them with the information from the roof, and then defines the value of each design issue about technical characteristics. The mathematical calculations and formal statements of the matrix can be found in section 6.2 in details. Any matrix may be added after another one, if the output of one matrix can be used as the input in the other one for further modelling such as product planning phase after design phase. The content of the matrices are redesigned according to the issue to be designed and the level of details, but same calculations are performed.

QFD evolved into an integrated methodology and reached its third generation which is called “Modern QFD” based on Blitz QFD (QFDI, 2006). Some new tools have been appended to its basic development process. In traditional QFD, the ordinal scale is used, which creates some risks about the acceptance by the scientist. In the Modern or third generation QFD, one type of ratio scale is preferred to perform mathematical and statistical operations.

Because QFD has a tailored analysis process and does not prescribe any scale, the relationships in the matrices are performed by a scale developed through the AHP’s pairwise comparison process to acquire user considerations to be used in the evaluations. QFD Institute of U.S. proposes AHP to obtain an original scale for each design process from the customers. In some parts of the matrices, different techniques can be integrated to reach more significant results. Recent QFD applications apply optimization or heuristic techniques to calculate the relationship values between customer requirements and technical characteristics. All relationships, as well as their significance rates, and priorities, conflicts (roofs of the houses) and benchmarking issues can be considered in an integrated manner through the series of the houses (matrices).

The QFD matrix is evaluated according to the relationships and benchmarking issues and the following operations are performed in each of the matrix. QFD performs sequential matrices where the results (weights of the characteristics given in columns) of any matrix is the inputs (rows in the first two columns) of the next matrix.

1) Priority of the inputs (part 7 in Figure 5.2): if it is the first matrix, it is obtained through AHP. Otherwise, the final normalized weights of the columns of the previous matrix are taken as the priorities of the next one.

2) The parts of the matrix (the parts in Figure 5.2) can also be weighted through the same techniques.

3) The adjusted importance weight (using part 1, 2 and 7 in Figure 5.2) for each row is then calculated through Equation (5.1) and (5.2):

4) $A_i = W_p P_i + W_c C_i$ (5.1)

$C_i = L_i / S_i$ (5.2)

Where

W_p: predefined weight of priority part.

P_i: priority of i^{th} row (obtained from the previous matrix or, e.g. AHP)

W_c: predefined weight of competitive improvement part

C_i: improvement ratio of i^{th} row

L_i: position plan for improvement of i^{th} row

S_i: current status of i^{th} row

5) Relationship values are assigned to part 4 with respect to the scales obtained using AHP.

6) The absolute importance weight for each column is calculated by Equation (5.3):

$A_j : \sum_{i=1}^{n} A_i R_{ij}$ (5.3)

Where

A_j: absolute importance weight of j^{th} column.

A_i: adjusted weight of i^{th} row.

R_{ij}: relationship value of entry ij.

7) The adjusted weights of the columns are calculated based on the absolute weights. The evaluations of part 6 in Figure 5.2 including the weights of functional characteristics, technical challenge and technical advantage, and their evaluation values obtained as a ratio scale through AHP are then calculated by Equation (5.4):

$AW_j = W_f F_j + W_{TA} TA_j + W_{TC} TC_j$ (5.4)

Where

W_f: predefined weight of functional characteristics part.

F_j: adjusted value of A_j after normalization in 0 and 1.

W_{TA}: predefined weight of technical advantage part.

TA_j: normalized value for technical advantage of the j^{th} column.

W_{TC}: predefined weight of technical challenge part.

TC_j: normalized value for technical challenge of the j^{th} column.

j: 1,2,…, n (n: total number of characteristics/columns to be evaluated in each matrix)

8) The roof of the matrix (part 5 in Figure 5.2) is considered only when the design targets are determined. Otherwise, it does not take part in the calculations.

9) The design targets are defined. The numerical values obtained in Equation (5.4) are transferred as priority values (P_i) for the next matrix.

All calculations in Modern QFD are performed using ratio-scale numbers. At the beginning of the project, scales for each part of the matrix are determined, e.g. for relationship matrix: no relationship, weak relationship, strong relationship, and very strong relationship; for benchmarking: none, low, medium, high, and very high. Then these scales are compared pairwise, i.e., with AHP, and then a ratio number is assigned for each scale defined for each evaluation. Then, according to the evaluations of customers, their ratio scale values are assigned to the corresponding matrix part. Finally, the calculations are performed following the equations from 5.1 to 5.4. As in the traditional QFD, weights of technical characteristics are obtained and design targets are determined. AHP calculations are explained in section 5.6 in details.

5.4 Software-QFD: S-Q(F)D

QFD is also modified for software development, called S-Q(F)D, through the modification of the house structures and integration with use cases. One of the earliest papers on applying QFD to software development was written by (Zultner, 1990). Zultner (1990) proposes a framework called Software Quality [Function] Deployment (SQD). This approach follows the idea of the deployment of the 'voice of the user' throughout the entire software development process. The problems associated with the development of software are a consequence of improperly defining customer requirements, SQD is an attempt to address this problem. The approach is not used in isolation but as a complementary framework to conventional software development approaches and project management techniques. The process is split into a number of phases. Each phase involves the production of a number of matrices describing the relationships between the various user and technical requirements, processes and entities. The starting point for SQD is the customer or "*the source of the voice – the user*". Table 5.8 represents the matrix structure in SQD phases.

Table 5.8 Matrix structure in SQD

❑ *Conceive → product planning*	Z-0 matrix – customer characteristics Vs customer segment Z-1 matrix – customer segment Vs customer voice A-1 matrix – customer voice Vs measurable system objectives A-2 matrix – measurable objectives Vs process and data models Decide: what is in the system
❑ *Develop → design planning*	A-4 matrix – measurable objectives Vs high level design (modules) E-0 technologies Vs concepts Technologies Vs high level design (modules) Decide: make or buy software modules, target machines….
❑ *Manufacture → process planning*	Perhaps configuration management system Perhaps inspection process
❑ *Deliver → production planning*	Perhaps roll-out plans Perhaps data migration plans Perhaps data quality measures

Reference: (Zultner, 1990)

The following steps describe the SQD process (Zultner, 1990):

1. User/user requirements

a) Identify users –The Z-0 matrix may be used in this task (see Table 5.8).

b) Determine user requirements – interviews, surveys, JAD or team analysis sessions, focus groups, trouble reports, problem logs. Compliments on any existing systems may be used to discover user needs and wants. These are then refined into concise user requirements statements and organised using affinity diagrams and relations diagrams into a final hierarchy of user requirements using the Z-1 matrix. Zultner (1990) recommends the usage of relations diagrams from successful practice.

c) *Prioritise user requirements* – user requirements are linked back to users, they are refined and their raw priorities are calculated using AHP.

d) *Adjust user requirements* – conflicts in user requirements always exist, or requirements are difficult to satisfy. Using an adjustment factor the raw priorities can be modified to focus on the most important ones. The adjustment factor can be the number of users in each category. What users want is reflected by the raw priorities, while the adjusted priorities reflect which users an organization wants to satisfy the most.

2. User requirements/technical requirements

a) *Determine technical requirements* – what technical characteristics contribute to or address user requirements in terms of processes and data. These are organised into hierarchies using AHP.

b) *Adjust user requirements* – consider "*the competitive position of the software and the sales points necessary for a user to buy the software*". This is followed by deploying the final user requirements weights.

c) *Prioritize technical requirements* – using the weights from the previous section, the linkages between user requirements and technical requirements are made.

3. Technical requirements/processes/entities

Technical requirements are deployed into software engineering models (Entity-Relationship diagrams and Data Flow diagrams) using the A-2 matrix in Table 5.8.

a) *Determine entities* – what data is required (from ERD).

b) *Determine processes* – what processing is required (from DFD).

4. Processes/entities:

Map processes to entities using the Z-2 matrix. Different matrices may be used to support different approaches, e.g. an object/entity/process matrix would support object orientation. Zultner relates user requirements to six software engineering models that can be deployed throughout the development lifecycle to deploy the user requirements. These are: DFD, Objective-hierarchy diagram, ER diagram, Control flow diagram, Event table and Access diagram. This approach requires a solid grounding in conventional software engineering, "*which provides the software specific engineering know-how and project management, which provides the planning and coordinating of time and resources necessary to build good software*" (Zultner, 1990).

Another one of the earliest published papers and case studies into the usage of QFD in software development was by (Betts, 1990). She recognises the areas where QFD is relevant to the product development process. The main idea is voice of the customer about the products customers will buy. QFD makes available methods address customer focused development such as the use of a cross functional team throughout the whole product life cycle, get the engineers out in the field and using affinity grouping for a deeper understanding of customer needs (Saeed, 2004).

Requirements engineering in general is an ongoing, social and interactive process, where human communication is one of the key issues if not the most important one. Furthermore, both authors view QFD as a way of involving users in the requirements engineering process and as a group session technique. These views are similar and correct, however, they overlook the other uses of QFD. Forming a cross-functional development team and involving users in the requirements engineering process is not what QFD is, rather it is a pre-requisite for almost any QFD project. QFD is a methodology for product development that incorporates the voice of the customer throughout the entire development cycle. Utilising the seven management tools and the matrices available, QFD has some facilities for organizing, prioritizing and documenting requirements. In addition, QFD enables the development team to trace back each proposed technical feature to the original voice of the customer. By recognising "the What-versus-How" dilemma (Hussein & Kremer, 2004), QFD allows for a clear separation between what is required and what can be done to address that requirement. Another capability of QFD is that it provides means to document technical targets and benchmarks that can be used to evaluate competition. Beyond the house of quality, more matrices can be built to document more detailed and varied information throughout the development process.

Zultner (1995) has declared that "Traditional software development is not focused. In order to satisfy the customer within the schedule and resource constraints that all projects face it is necessary to concentrate the best efforts on those things of greatest importance to the stakeholders of the system". The idea is that by deploying the most important customer requirements throughout the entire development process, focused and coherent software development which leads to great software can be achieved. On the other hand, by only deploying the voice of the customer at the start, the developed software may have medium quality. Zultner (1995) also defines QFD as "a method for focusing the effort and limited resources of a project team on what delivers the best value to the most important stakeholders" and SQFD as "a technique that is specifically developed for the use of QFD-techniques in information systems development" (Zultner, 1995). The SQFD method is available commercially as part of the Anderson Consulting (ACCENTURE) Method/1 (Saeed, 2004). The SQFD process in Method/1 is described in six steps:

1. Determine stakeholder types and characteristics.
2. Evaluate stakeholder inputs
3. Define business needs
4. Assign business needs to stakeholder types.
5. Align requirements to needs
6. Manage value

Richard Zultner is one of the authors of the earliest published papers on the topic of QFD and software development. Later on, he modified his Software QFD approach to a subset of QFD designed to provide the maximum gains from minimum effort. This new approach was called Blitz QFD (Zultner, 1997). To do Blitz QFD it is argued that it requires exploration (what are the most important things we need to know to satisfy the customer) and execution (what are the most important things for us to do to deliver value to the customers). Zultner (2000) argues that doing QFD does not necessitate the development of matrices and that the issue is not about a complete set of requirements but a sufficient one (the smallest subset) to satisfy the customer needs. Figure 5.10 and Figure 5.11 includes the detailed steps of modified Software-QFD as a roadmap and process, respectively.

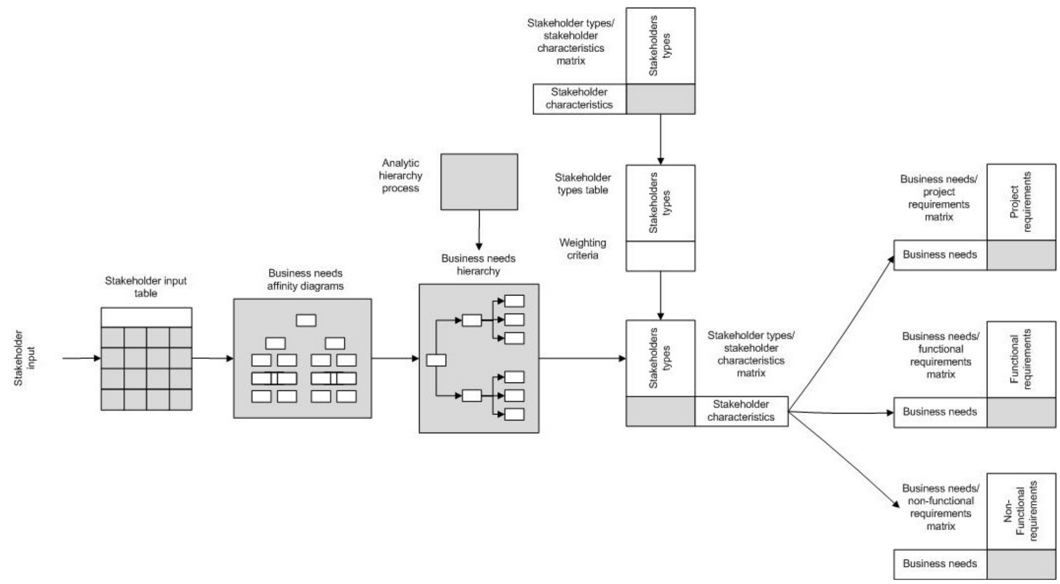


Figure 5.10 Software-QFD road map.
Reference: (Saeed, 2004)

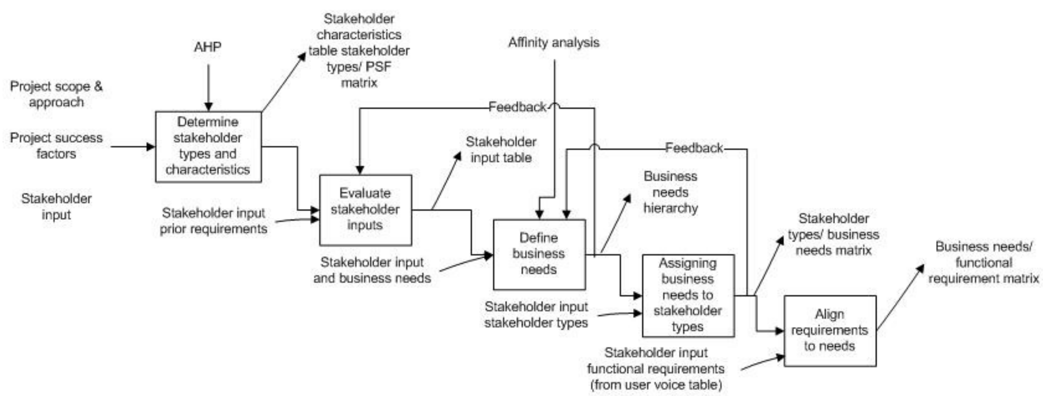


Figure 5.11 Software-QFD process.
Reference: (Saeed, 2004)

Since QFD has its roots in manufacturing industry, the product characteristics in the QFD matrix originally correspond to measurable quality characteristics. Yet software is identified not by its physical characteristics but by its behaviour, so the construction of the house names and contents are used in a different way. Quality matrix structure of SQFD is represented in Figure 5.12.

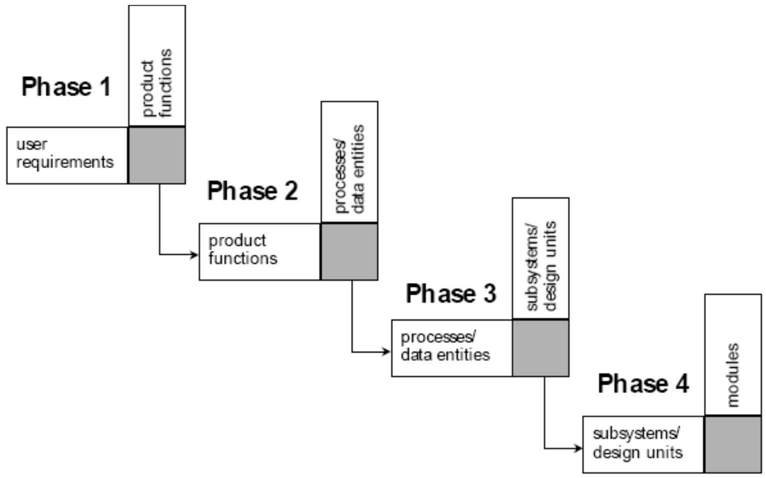


Figure 5.12 Zultner's S-QFD structure.
Reference: (Herzwurm, Schokert, Dowie, & Breidung, 2002)

The matrix flow of software-QFD includes four phases. The first phase converts user requirements into product functions, and the second phase converts product functions into processes and data analysis. Then the third and fourth phase defines sub systems and designs modules for the required application (Herzwurm, Schokert, Dowie, & Breidung, 2002).

Recall from section 4.8 that Joint Application Development (JAD) is a facilitated multi-disciplinary group meeting. This meeting or set of meetings are designed for all the relevant system stakeholders (IT personnel, users, management etc.) to get together to discuss system requirements (Avison & Fitzgerald, 2003). JAD is simply a group meeting with a set of designated roles that each party in the meeting plays. JAD does not provide any tools or techniques that are unique to the methodology. It does not describe how to organise, prioritise and document customer needs and technical characteristics of a system. With JAD the initiating point for the project is management and management wants. On the other hand, QFD is a methodology to product development with greater scope than JAD (Brown, 2004).

The initiating point for QFD is management but not what management needs, rather, what customers need. QFD and JAD are similar techniques in that they both look to involve stakeholders in the development process. The focus of QFD is customer needs and quality, while JAD is more concerned with the interaction between the various stakeholders in meetings. Both QFD and JAD are general techniques and can be used in any industry. With JAD the only difference would be the designation of roles in various industries. QFD and JAD can complement one another: QFD is the tool for documenting and managing the requirements while JAD is the way meetings are organised and structured. QFD is a general product planning and decision making tool. Any phase or stage that involves planning and decision making can be addressed by QFD. Even though customer needs and technical responses are the primary focus of the house of quality matrix, for example, any set of interrelating criteria can also be evaluated using QFD matrices. The strategy, feasibility and analysis phases cover the overall requirements analysis tasks while the evaluation phase attempts to ensure that customer needs are met. These areas are directly addressed by QFD. Regarding the maintenance phase, the use of QFD matrices to compare various criteria can be done to help planning (Brown, 2004).

S-QFD is a common requirements modelling tool for software developers and has many applications samples in recent years, e.g. see (Büyüközkan & Feyzioğlu, 2005); (Elboushi & Sherif, 1997); (Lesley, 2000) .

Especially, S-QFD method is developed by modifying the house of quality matrices with respect to the requirement structures in software engineering. These publications represent QFD based analysis and standardization for requirements analysis of software development (ISO 9126) (Zrymiak, 2003).

In QFD, however, these ambiguous sounding quality requirements eventually evolve into very technical, non-ambiguous requirements. It is all part of the process. The experience is that a software project is typically driven by a combination of factors, only some of which are customer needs. For example, it is seen that projects where the prime objective is to explore technology, the company does not understand well as a way to increase understanding of users and minimize risk in the long term. On the other hand, if the developers of a project are strictly limited to making the customer happy, the developers will choose QFD as a standard for "voice of the customer" (Denney, 2005).

As a summary, by the viewpoint of software projects and/or requirements engineering, QFD is a systematic planning tool and can be used for software as a product, but with different structure and matrix flow. So, the QFD methodology constructed for software development is called software-QFD or S-QFD. S-QFD is examined in this study because of the similarity of requirements analysis for software development and enterprise modelling. The proposed approach presented in chapter 6 is a requirements analysis methodology developed for enterprise modelling. This approach involves some modifications in standard QFD methodology with respect to the different needs in enterprise modelling. Enterprise-QFD and S-QFD have similarities during the development process but Enterprise-QFD differentiates in details.

5.5 Optimization Studies in QFD

Traditional QFD methodology uses ordinal numbers for its matrices. But in some part of matrices, different techniques can be integrated to reach more significant results. For example, AHP, Fuzzy-AHP, and ANP techniques are used in the literature to prioritize customer requirements. Recent QFD applications try to apply optimization or heuristic techniques to calculate the relationship values between customer requirements and technical characteristics. Fuzzy logic and Fuzzy regression models are also very popular in this field of research. Optimization techniques can be also applied to these regression models. Among the studies in literature, genetic algorithms, neural networks and their fuzzy modelling approaches are frequently applied in QFD studies. Some examples from the literature can be presented as (Tang, Fung, Xu, & Wang, 2002), (Karsak, 2004), (Chen & Weng, 2003), (Yang, Wang, Dulaimi, & Low, 2003), (Kim, Moskowitz, Dhingrra, & Evans, 2000), (Temponi, Yen, & Tiao, 1999), (Zhou, 1998), (Myint, 2003), (Vairaktarakis, 1999), (Park & Kim, 1998), (Markovitz & Kim, 1997), (Karsak, Sözer, & Alptekin, 2002), (Bai & Kwong, 2003), (Liu, 2005), (Lee & Kusiak, 2001), (Chen & Chen, 2005).

5.6 Analytic Hierarchy Process and Related Techniques

5.6.1 The AHP Method

In daily lives, people often have to make decisions. People learn by trying and by example. Deciding too quickly can be hazardous; delaying too long can mean missed opportunities. In the end, it is crucial that people make up their mind. What people need is a systematic and comprehensive approach to decision making (Saaty, 2001).

In evaluating n competing alternatives $A_1, \ldots, A_n$ under a given criterion, it is natural to use the framework of pairwise comparisons represented by an $n \times n$ square matrix from which a set of preference values for the alternatives is derived. Many methods for estimating the preference values from the pairwise comparison matrix have been proposed and their effectiveness comparatively evaluated. Some of the proposed estimating methods presume interval-scaled preference values. But most of the estimating methods proposed and studied are within the paradigm of the analytic hierarchy process that presumes ratio-scaled preference values. The main challenge is how to reconcile the inevitable inconsistency of the pairwise comparison matrix elicited from the decision makers in real-world applications. When the decision maker is unable to rank the alternatives holistically and directly with respect to a criterion, pairwise comparisons are often used as intermediate decision support (Choo & Wedley, 2004).

In this part, the analytical way to make decisions in the status of multiple and multi-level criteria is presented with the viewpoint of AHP approach. The analytic hierarchy process (AHP) is a method for ranking decision alternatives and selecting the best one when the decision maker has multiple criteria (Taylor, 2004). It answers the question "Which one?". The decision maker will select the alternative that best meets his or her decision criteria. AHP is a process for developing a numerical score to rank each decision alternative based on how well each alternative meets the decision maker's criteria (Russell & Taylor, 2003).

In AHP, preferences between alternatives are determined by making pairwise comparisons. In a pairwise comparison the decision maker examines two alternatives by considering one criterion and indicates a preference. These comparisons are made using a preference scale, which assigns numerical values to different levels of preference (Saaty, 1992). The standard preference scale used for AHP is 1-9 scale which lies between “equal importance” to “extreme importance”.

The standard preference scale used for AHP is shown in Table 5.9.

Table 5.9 The fundamental AHP scale

Value	Definition	Details
1	Equal Importance	Two activities contribute equally to the objective
2	Weak	
3	Moderate importance	Experience and judgment slightly favour one activity over another
4	Moderate plus	
5	Strong importance	Experience and judgment strongly favour one activity over another
6	Strong plus	
7	Very strong or demonstrated importance	An activity is favoured very strongly over another; its dominance demonstrated in practice
8	Very, very strong	
9	Extreme importance	The evidence favouring one activity over another is of the highest possible order of affirmation

Reference: (Saaty, 1992, p.6)

As seen in Table 5.9, the verbal terms of the Saaty’s fundamental scale of 1–9 is used to assess the intensity of preference between two elements. The value of 1 indicates equal importance, 3 moderately more, 5 strongly more, and 7 very strongly and 9 extremely more importance, respectively. The values of 2, 4, 6, and 8 are allotted to indicate compromise values of importance. In the pairwise comparison matrix, value 9 indicates that one factor is extremely more important than the other, and value 1/9 indicates that one factor is extremely less important than the other, and value 1 indicates equal importance. Also, if the importance of one factor with respect to a second is given, then the importance of the second factor with respect to the first is the reciprocal. This means a_{ij}=9 => a_{ji}=1/9. Ratio scale and the use of verbal comparisons are used for weighting of quantifiable and non-quantifiable elements (Pohekar & Ramachandran, 2004).

In AHP, for an elaborate design the following process should be applied (Saaty, 1992).

1. Identify the overall goal. What is the main question?
2. Identify the sub goals of the overall goal. If relevant, identify time horizons that affect the decision.
3. Identify the criteria that must be satisfied to fulfil the sub goals of the overall goal.
4. Identify sub criteria under each criterion. Note that criteria or sub criteria may be specified in terms of ranges of values of parameters or in terms of verbal intensities such as high, medium, low.
5. Identify the actors involved.
6. Identify the actor goals.
7. Identify the actor policies.
8. Identify the options or outcomes.
9. For yes-no decisions take the most preferred outcome and compare benefits and costs of making the decision with those of not making it.
10. Do benefit/cost analysis using marginal values. Ask which alternative yields the greatest benefit; for costs, which alternative costs the most. Proceed similarly if a risks hierarchy is included.

AHP has been applied in a variety of contexts: from the simple everyday problem of selecting a school to the complex problems of designing alternative future outcomes of a developing country, evaluating political candidacy, allocating energy resources, and so on. AHP enables decision-makers to structure an unstructured complex problem in the form of a simple hierarchy and also provides a decision platform to evaluate a large number of quantitative and qualitative factors in a systematic manner where the criteria set has some conflicts. The application of AHP to complex problems usually involves four major steps (Cheng, Yang, & Hwang, 1999):

1. Break down the complex problem into a number of small constituent elements and then structure the elements in a hierarchical form.
2. Make a series of pairwise comparisons among the elements according to a ratio scale 1, 3, 5, 7 and 9.
3. Use the eigenvalue method to estimate the relative weights of the elements.
4. Aggregate these relative weights and synthesize them for the final measurement of given decision alternatives.

AHP is a powerful and flexible multi-criteria decision-making tool for dealing with complex problems where both qualitative and quantitative aspects need to be considered. AHP helps analysts to organize the critical aspects of a problem into a hierarchy rather like a family tree. By reducing complex decisions to a series of simple comparisons and rankings, then synthesizing the results, AHP not only helps analysts to arrive at the best decision, but also provides a clear rationale for the choices that are made (Bevilacqua, D'Amore, & Polonora, 2004).

The essence of the process is decomposition of a complex problem into a hierarchy with goal (objective) at the top of the hierarchy, criteria and sub-criteria at levels and sub-levels of the hierarchy, and decision alternatives at the bottom of the hierarchy. Elements at given hierarchy levels are compared in pairs to assess their relative preference with respect to each of the elements at the next higher level. The method computes and aggregates their eigenvectors until the composite final vector of weight coefficients for alternatives is obtained. The entries of final weight coefficients vector reflect the relative importance (value) of each alternative with respect to the goal stated at the top of the hierarchy (Pohekar & Ramachandran, 2004). A decision maker may use this vector according to his particular needs and interests. To elicit pairwise comparisons performed at a given level, a matrix *A* is created in turn by putting the result of pairwise comparison of element *i* with element *j* into the position a_{ji} as below.

$$
A = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ . \\ C_n \end{array}
\begin{array}{c} \begin{array}{cccccccc} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & . & C_n \end{array} \\
\begin{bmatrix}
1 & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & . & a_{1n} \\
a_{21} & 1 & a_{23} & a_{24} & a_{25} & a_{26} & . & a_{2n} \\
a_{31} & a_{32} & 1 & a_{34} & a_{35} & a_{36} & . & a_{3n} \\
a_{41} & a_{42} & a_{43} & 1 & a_{45} & a_{46} & . & a_{4n} \\
a_{51} & a_{52} & a_{53} & a_{54} & 1 & a_{56} & . & a_{5n} \\
a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 1 & . & a_{6n} \\
. & . & . & . & . & . & 1 & . \\
a_{n1} & a_{n2} & a_{n3} & a_{n4} & a_{21} & a_{21} & . & 1
\end{bmatrix} \end{array}
$$

Where

n = the number of criteria to be evaluated

$C_i = i^{th}$ criterion,

A_{ij} = importance of i^{th} criterion according to j^{th} criterion.

After obtaining the weight vector, it is then multiplied with the weight coefficient of the element at a higher level (that was used as criterion for pairwise comparisons). The procedure is repeated upward for each level, until the top of the hierarchy is reached. The overall weight coefficient, with respect to the goal for each decision alternative is then obtained. The alternative with the highest weight coefficient value should be taken as the best alternative (Saaty & Vargas, 1994).

5.6.2 Application of AHP Method

In this part, the AHP methodology is explained in more detail. A hierarchical structure with respect to the methodology is shown in Figure 5.13. The values in the pairwise comparison matrix are as follows.

P_{ij}: Relative importance level of i^{th} alternative or criterion according to j^{th} alternative or criterion.

W_{ik}: Relative importance level *of* i^{th} alternative or criterion to k^{th} alternative or criterion.

C_k: k^{th} alternative or criterion.

WS_{ik}: Weighted sum of i^{th} alternative or criterion with respect to k^{th} alternative or criterion.

n: The number of alternatives.

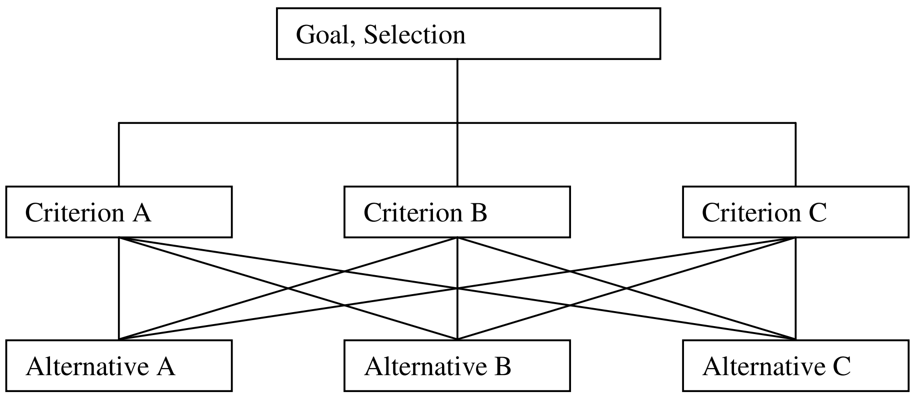


Figure 5.13 General AHP structure.

Symbolic exhibition of the pairwise comparison matrix that shows the priority of alternatives from the viewpoint of the first criterion has been shown in Table 5.10.

Table 5.10 Symbolic exhibition of the pairwise comparison matrix that shows the priority of alternatives

C_1	Alternative 1	Alternative 2	Alternative 3
Alternative 1	P_{11}	P_{12}	P_{13}
Alternative 2	P_{21}	P_{22}	P_{23}
Alternative 3	P_{31}	P_{32}	P_{33}

The determination of the importance levels according to the data in Table 5.10 has been explained step by step as follows:

Step 1. Sum of the values in each column, which is in Table 5.11.

Table 5.11 The calculation of the sum of the columns

$\sum_{i=1}^{n} P_{i1}$	$\sum_{i=1}^{n} P_{i2}$	...	$\sum_{i=1}^{n} P_{in}$

Step 2. Divide each element in the pairwise comparison matrix by the sum of the columns in question. The exhibition of the operations made in Step 2 is in Table 5.12.

Table 5.12 Division of the elements by the sum of the columns

Criterion	Alternative 1	Alternative 2	...	Alternative n
Alternative 1	$\frac{P_{11}}{\sum_{i=1}^{n} P_{i1}}$	$\frac{P_{12}}{\sum_{i=1}^{n} P_{i2}}$	...	$\frac{P_{1n}}{\sum_{i=1}^{n} P_{in}}$
Alternative 2	$\frac{P_{21}}{\sum_{i=1}^{n} P_{i1}}$	$\frac{P_{22}}{\sum_{i=1}^{n} P_{i2}}$	...	$\frac{P_{2n}}{\sum_{i=1}^{n} P_{in}}$
...	...	...	...	...
Alternative n	$\frac{P_{n1}}{\sum_{i=1}^{n} P_{i1}}$	$\frac{P_{n2}}{\sum_{i=1}^{n} P_{i2}}$	...	$\frac{P_{nn}}{\sum_{i=1}^{n} P_{in}}$

Note: The sum of each columns should be equal to 1.

Step 3. Calculate the average of the elements in each column. The importance levels found in the result of this operation are in Table 5.13.

Table 5.13 The importance levels

Criterion	The Importance Level
Alternative 1	$\frac{\frac{P_{11}}{\sum_{i=1}^{n} P_{i1}} + \frac{P_{12}}{\sum_{i=1}^{n} P_{i2}} + ... + \frac{P_{1n}}{\sum_{i=1}^{n} P_{in}}}{n} = W_{11}$
Alternative 2	$\frac{\frac{P_{21}}{\sum_{i=1}^{n} P_{i1}} + \frac{P_{22}}{\sum_{i=1}^{n} P_{i2}} + ... + \frac{P_{2n}}{\sum_{i=1}^{n} P_{in}}}{n} = W_{21}$
...	...
Alternative n	$\frac{\frac{P_{n1}}{\sum_{i=1}^{n} P_{i1}} + \frac{P_{n2}}{\sum_{i=1}^{n} P_{i2}} + ... + \frac{P_{nn}}{\sum_{i=1}^{n} P_{in}}}{n} = W_{n1}$

These data have shown the relative importance levels of three alternatives from the point of view of the first criterion as percentage. These data can be written as a priority vector.

$$W_{i1} = \begin{bmatrix} W_{11} \\ W_{21} \\ \ldots \\ W_{n1} \end{bmatrix} \qquad (i = 1, 2, \ldots, n)$$

The procedure used so as to calculate the priorities for all decision alternatives is the same as step 1, 2 and 3. The weighted point of each criterion, i.e., the priority level, has been multiplied with the priority level of the alternative which has been compared according to this criterion. This operation has been reiterated for all criteria. The importance level of the alternative when all criteria have been taken into consideration has been found after the values have been summed.

$$\sum_{i=1}^{m} W_{c_i}.W_{1i} = W_1$$

$$\sum_{i=1}^{m} W_{c_2}.W_{2i} = W_2$$

....

$$\sum_{i=1}^{m} W_{c_n}.W_{ni} = W_n$$

The best alternative = maximum *(W_1, $W_{2,}$...$W_{n,}$)*

5.6.3 Fuzzy-AHP

In the classical AHP, the decision maker is asked to supply exact pairwise comparison ratios r_{ij} between sub-criteria $A_1; \ldots ; A_n$ for each criterion in each level of the hierarchy (Saaty & Vargas, 1994). These comparison ratios form the comparison matrix whose principal eigenvector gives the relative weights of the sub-criteria. There is an extensive literature that addresses the situation where the comparison ratios are imprecise judgments (Leung & Chao, 2000). In most of the real-world problems, some of the decision data can be precisely assessed while others cannot. Humans are unsuccessful in making quantitative predictions, whereas they are comparatively efficient in qualitative forecasting (Kulak & Kahraman, 2005). Essentially, the uncertainty in the preference judgments gives rise to uncertainty in the ranking of alternatives as well as difficulty in determining consistency of preferences. These applications are performed with respect to many different perspectives and extensions for fuzzification of AHP.. In this thesis study, extended analysis on Fuzzy-AHP is formulated for a selection problem (Chang, 1992).

The Fuzzy-AHP technique can be viewed as an advanced analytical method developed from the traditional AHP. Despite the convenience of AHP in handling both quantitative and qualitative criteria of multi-criteria decision making problems based on decision makers' judgments, fuzziness and vagueness existing in many decision-making problems may contribute to the imprecise judgments of decision makers in conventional AHP approaches (Bouyssou, Marchant, Pirlot, Perny, Tsoukias, & Vincke, 2000). So, many researchers, e.g., (Boender, De Graan, & Lootsma, 1989); (Buckley, 1985); (Laarhoven & Pedrycz, 1983); (Lootsma, 1997); (Riberio, 1996), who have studied Fuzzy-AHP which is the extension of Saaty's theory, have provided evidence that Fuzzy-AHP shows relatively more sufficient description of these kinds of decision making processes compared to the traditional AHP methods. Yu (2002) employed the property of goal programming to solve group decision-making Fuzzy-AHP problems. Sheu (2004) presented a fuzzy-based approach to identify global logistics strategies. Kulak & Kahraman (2005) used Fuzzy-AHP for multi-criterion selection among transportation companies. Kuo, Chi, & Kao (2002) integrated Fuzzy-AHP and artificial neural networks for selecting convenience store location. Cheng (1996) proposed a new algorithm for evaluating naval tactical missile systems by Fuzzy-AHP based on grade value of membership function. Zhu, Jing, & Chang (1999) and Chang (1996) made a discussion on the extended analysis method and applications of Fuzzy-AHP.

In complex systems, the experiences and judgments of humans are represented by linguistic and vague patterns. Therefore, a much better representation of this linguistics can be developed as quantitative data. This type of data set is then refined by the evaluation methods of fuzzy set theory. On the other hand, the AHP method is mainly used in nearly crisp (non-fuzzy) decision applications and creates and deals with a very unbalanced scale of judgment. Therefore, the AHP method does not take into account the uncertainty associated with the mapping. The AHP's subjective judgment, selection and preference of decision-makers have great influence on the success of the method. The conventional AHP still cannot reflect the human thinking style. Avoiding these risks on performance, Fuzzy-AHP was developed to solve the hierarchical fuzzy problems (Cheng, Yang, & Hwang, 1999).

Chang's extent analysis on Fuzzy-AHP depends on the degree of possibilities of each criterion. According to the responses on the question form, the corresponding triangular fuzzy values for the linguistic variables are placed and for a particular level on the hierarchy the pairwise comparison matrix is constructed. Sub totals are calculated for each row of the matrix and new *(l, m, u)* set is obtained. Then, in order to find the overall triangular fuzzy values for each criterion, $l_i/\Sigma l_i$, $m_i/\Sigma m_i$, $u_i/\Sigma u_i$, $(i=1,2,..., n)$ values are found and used as the latest $M_i(l_i,m_i,u_i)$ set for criterion M_i in the rest of the process. In the next step, membership functions are constructed for each criterion and intersections are determined by comparing each couple. In the fuzzy logic approach, the intersection point is found for each comparison, and the membership values of the point correspond to the weight of that point. Each membership value can also be defined as the degree of possibility. For a particular criterion, the minimum degree of possibility of the situations where the value is greater than the others is also the weight of this criterion before normalization. After obtaining the weights for each criterion, they are normalized and called the final importance degrees or weights for the hierarchy level (Chang, 1996).To apply the Fuzzy-AHP process, according to the method of Chang's extent analysis in (Chang, 1996) , each criterion is taken into account for extent analysis. In extent analysis, $g_i$is performed for each criterion. Therefore, *m extent* analysis values for each criterion can be obtained by using the following notation (Kahraman, Cebeci, & Ruan, 2004):

$$M_{g_i}^1, M_{g_i}^2, M_{g_i}^3, M_{g_i}^4, M_{g_i}^5, ..., M_{g_i}^m$$

where g_i is the goal set *(i = 1, 2, 3, 4, 5,...,n)* and all $M_{g_i}^{j}$ *(j = 1, 2,..., m)* are Triangular Fuzzy Numbers *(TFNs)*. The steps of Chang's analysis can be given as in the following:

Step 1: The fuzzy synthetic extent value *(S_i)* with respect to the i^{th} criterion is defined by equation 5.5, 5.6 and 5.7. Fuzzy addition operation of *m* extent analysis values for a particular matrix is employed. At the end step of the calculation, new *(l,m,u)* set is obtained and used for the next phase.

$$S_i = \sum_{j=1}^{m} M_{g_i}^{j} \otimes [\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j}]^{-1} \quad (5.5)$$

$$\sum_{j=i}^{m} M_{g_i}^{j} \quad (5.6)$$

$$\sum_{j=1}^{m} M_{g_i}^{j} = (\sum_{j=1}^{m} l_j, \sum_{j=1}^{m} m_j, \sum_{j=1}^{m} u_j) \quad (5.7)$$

Where *l* is the lower limit value, *m* is the most promising value and *u* is the upper limit value. The fuzzy addition operation for $M_{g_i}^{j}$ *(j = 1, 2,..., m)* are performed with respect to the equation 5.8, inverse operation is employed to compute the inverse of the vector in equation 5.9 by using equation 5.10.

$$\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j} = (\sum_{i=1}^{n} l_i, \sum_{i=1}^{n} m_i, \sum_{i=1}^{n} u_i) \quad (5.8)$$

$$[\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j}]^{-1} \quad (5.9)$$

$$[\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j}]^{-1} = \left(\frac{1}{\sum_{i=1}^{n} u_i}, \frac{1}{\sum_{i=1}^{n} m_i}, \frac{1}{\sum_{i=1}^{n} l_i} \right) \quad (5.10)$$

Step 2: The degree of possibility $M_2 = (l_2, m_2, u_2) \geq M_1 = (l_1, m_1, u_1)$ is defined by equation 5.11:

$$V(M_2 \geq M_1) = \sup_{y \geq x} [\min(\mu_{M_1}(x), \mu_{M_2}(y))] \quad (5.11)$$

where x and y are the values on the axis of membership function of each criterion. This expression can be equivalently written as in equation 5.12 below:

$$V(M_2 \geq M_1) = \begin{cases} 1, & \textit{if } m_2 \geq m_1, \\ 0, & \textit{if } l_1 \geq u_2, \\ \dfrac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)} & \textit{otherwise} \end{cases} \qquad (5.12)$$

where d is the highest intersection point μ_{M_1} and μ_{M_2} (see Figure 5.14) (Zhu et al, 1999: 451).

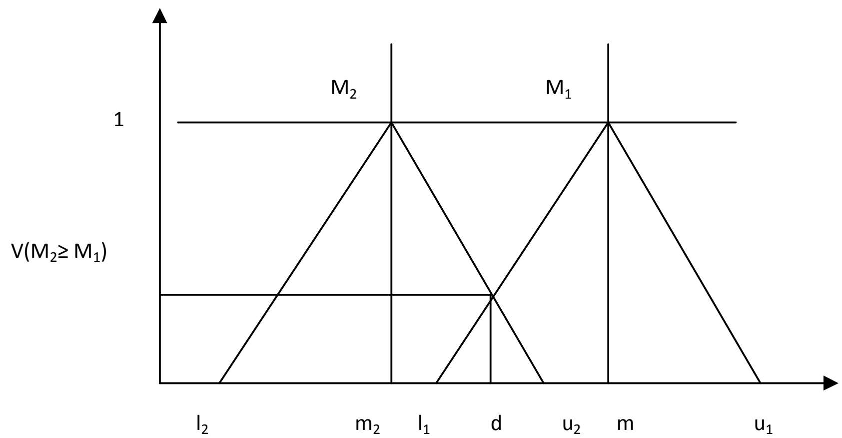


Figure 5.14 The intersection between M_1 and M_2.
Reference: (Zhu, Jing, & Chang, 1999, p. 452)

To compare M_1 and M_2, we need both the values of $V(M_2 \geq M_1)$ and $V(M_1 \geq M_2)$:

Step 3. The degree possibility for a convex fuzzy number to be greater than k convex fuzzy numbers M_i $(i = 1, 2,..., k)$ can be defined by

$V(M \geq M_1, M_2, M_3, M_4, M_5, M_6, ..., M_k) =$

$V[(M \geq M_1)$ and $(M \geq M_2)$ and $(M \geq M_3)$ and $(M \geq M_4)$ and $(M \geq M_k)] =$

$\min V(M \geq M_i), i = 1, 2,..., k.$

Assume that $d'(A_i)$ in equation 5.13 is defined as follows:

$$d'(A_i) = \min V(S_i \geq S_k) \qquad (5.13)$$

For $k = 1, 2, ..., n; k \neq i$. Then the weight vector is given by equation 5.14:

$$W' = (d'(A_1), d'(A_2), d'(A_3), d'(A_4), d'(A_5),..., d'(A_n))^T \qquad (5.14)$$

Where A_i $(i = 1, 2,..., n)$ are n elements.

Step 4. Via normalization, the normalized weight vectors are determined as in equation 5.15:

$W = (d(A_1), d(A_2), d(A_3), d(A_4), d(A_5), d(A_6), ..., d(A_n))$ (5.15)

Where W is non-fuzzy numbers.

To evaluate the questions, people only select the related linguistic variable, then for calculations, they are converted to the following scale including triangular fuzzy numbers developed by (Chang, 1996) and generalized for such analysis as given in Table 5.14 below:

Table 5.14 TFN values

Statement	**TFN**
Absolute	(7/2, 4, 9/2)
Very strong	(5/2, 3, 7/2)
Fairly strong	(3/2, 2, 5/2)
Weak	(2/3, 1, 3/2)
Equal	(1, 1, 1)

Reference: (Tolga, Demircan, & Kahraman, 2005)

5.6.4 ANP and Fuzzy ANP

ANP is a more general form of AHP. Whereas AHP models a decision making framework using a uni-directional hierarchical relationship among decision levels, ANP allows for more complex interrelationships among the decision levels and components (Saaty, 2001). Typically, in AHP the top element of the hierarchy is the overall goal for the decision model. The hierarchy decomposes from a general to a more specific attribute until a level of manageable decision criteria is met. ANP does not require this strictly hierarchical structure. Interdependencies may be graphically represented by two way arrows (or arcs) among levels, or if within the same level of analysis, a looped arc. The directions of the arcs, in this case, signify dependence, arcs emanate from an attribute to other criteria that may influence it. The relative importance or strength of the impacts on a given element is measured on a ratio scale similar to AHP. A priority (relative importance weighting) vector may be determined by asking the decision maker for their numerical weight directly, but there may be less consistency, since part of the process of decomposing the hierarchy is to provide better definitions of higher level criteria (Sarkis, 1998).

ANP problem formulation starts by modelling the problem that depicts the dependence and influences of the factors involved to the goal or higher-level performance objective. These dependence and influences are subjectively judged by pairwise comparisons (Tesfamariam & Lindberg, 2005). The ANP approach is capable of handling interdependence among elements by obtaining the composite weights through the development of a 'supermatrix' (Sarkis, 1998). A supermatrix is constructed whose columns are the vectors as found in the earlier step. Different ways of manipulations of the supermatrix based on the particular type of the problem formulation results the limiting weights of the criteria.

Step 1. Model Construction and Problem Structuring: The first step is to construct a model to be evaluated. The model development will require the delineation of criteria at each level and a definition of their relationships.

Step 2. Pairwise Comparisons Matrices of lnterdependent Component Levels: Eliciting preferences of various components and criteria will require a series of pairwise comparisons where the decision maker will compare two components at a time with respect to an upper level 'control' criterion. These comparisons are collected in a pairwise comparison matrix. In ANP, like AHP, pairwise comparisons of the elements in each level are conducted with respect to their relative importance towards their control criterion (Sarkis & Talluri, 2004).

Step 3. Supermatrix Formation: The supermatrix allows for a resolution of the effects of interdependence that exists between the elements of the ANP network. The supermatrix is a partitioned matrix, where each sub-matrix is composed of the pairwise comparison matrices formed in Step 2 or some of are zero sub-matrices (all the elements in a zero sub-matrix are zero).

Step 4. Analyze sub-components: A similar pairwise comparison that was made in Step 2 is made for the criteria level for relative importance weight calculation (or eigenvector determination).

Step 5. Alternative Program, Project, or Technology Evaluations: Each alternative will need to be evaluated on each of the sub-criteria. This evaluation is completed by making a pairwise comparison of the performance or impact of each alternative on each sub- criteria.

Step 6. Selection of Best Alternative: The selection of the best alternative depends on the calculation of the 'desirability index' for an alternative *i*.

As seen in ANP methodology, ANP requires AHP method for its sub matrices where Fuzzy ANP requires Fuzzy-AHP evaluations for the sub processes and matrices following the steps that are explained in sections 5.6.2 and 5.6.3.

Fuzzy ANP is relatively new approach developed from AHP and Fuzzy-AHP methods. A few studies can be found on Fuzzy ANP models in the literature. The Fuzzy ANP method was first proposed by (Mikhailov & Singh, 2003). A short communication in Fuzzy ANP has been made in (Yu & Cheng, 2007). An integrated Fuzzy ANP approach to formulate and solve a QFD problem has been employed by (Kahraman, Ertay, & Büyüközkan, 2006). The ANP method deals only with crisp comparison ratios. However, uncertain human judgments with internal inconsistency obstructing the direct application of the ANP are frequently available.

5.6.5 Realizing the Operations Related to the ANP Methodology.

Common steps are applied for both ANP and Fuzzy ANP after the pairwise comparisons are completed for all related criteria. Following steps are applied to obtain final important values or weights of alternatives:

Step 1. The importance levels in the network which is calculated with the Fuzzy – AHP/AHP are taken as column matrices. A supermatrix (S_{ij}) is then obtained from conjunction of these matrices (W_i).

$$S = \begin{bmatrix} W_{11} & W_{12} & \ldots\ldots & W_{1m} \\ W_{21} & W_{22} & \ldots\ldots & W_{2m} \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots \\ W_{n1} & W_{n2} & \ldots\ldots & W_{nm} \end{bmatrix}$$

Step 2. Column totals are controlled. Column totals should be equal to 1. If the column totals ($\sum_{i=1}^{n} W_{i1}, \sum_{i=1}^{n} W_{i2}, \sum_{i=1}^{n} W_{i3}, \ldots\ldots \sum_{i=1}^{n} W_{im}$) are not equal to 1, they are equalized to 1 via normalization. Therefore, the weighted supermatrix has been provided.

Step 3. The weighted supermatrix is considered a Markov transition matrix. This matrix is used to find the steady state matrix, which is called converged supermatrix in ANP methodology. So, the matrix (S^N) showing the last importance levels for main criteria is calculated. In this matrix, all values in a column are equal.

$$S^N = \prod_{i=1}^{N} S \quad (N=1,2,\ldots\infty) \tag{5.16}$$

Finally, corresponding weights of importance or scales, or the target weight value of a particular selection problem are obtained using equation 5.16 because of its complexity level.

CHAPTER SIX
ENTERPRISE REQUIREMENTS ANALYSIS THROUGH QFD: ENTERPRISE-QFD

6.1 Introduction

Enterprises are managed as integrated systems. They deploy their strategies to perform their operations in order to achieve their goals. Enterprise models are the road maps to manage the enterprise with respect to the predefined goals and subsystems within an enterprise. These subsystems are established for different purposes, e.g., production management systems for planning efficient operations, quality systems for systematic management, environmental management systems, occupational safety systems, product development systems, financial management and cost systems. However, enterprises can survive if they are managed as a whole. This necessity forces enterprise managers to consider a common model providing the required infrastructure for each subsystem mentioned above. This modelling level is called enterprise modelling which has predefined stages starting with enterprise goals and objectives and ending when the enterprise applications are developed. A structure of an enterprise and modelling concept are given in Figure 2.7 in section 2.11 to support this consideration and the same figure is represented as Figure 6.1 to emphasize the situation.

Figure 6.1 emphasizes that the enterprise model should be well-matched with the enterprise goals and objectives. These goals and objectives cover not only the goals of the top management but also the goals of all entities interacting with the enterprise. Figure 6.1 also implies that successful enterprise application and software integration needs a model driven management, which is provided by the enterprise model. That is, the integration of all systems within an enterprise application requires an enterprise model that in turn needs a requirements model and analysis based on the enterprise goals and objectives. Enterprise integration is generally handled from different perspectives. Therefore, a specific framework comes in view to integrate these perspectives for a complete integration process.

Figure 6.1 is a common framework gathering organizational definitions, e.g. vision, mission, processes, and architectural and technical perspectives.

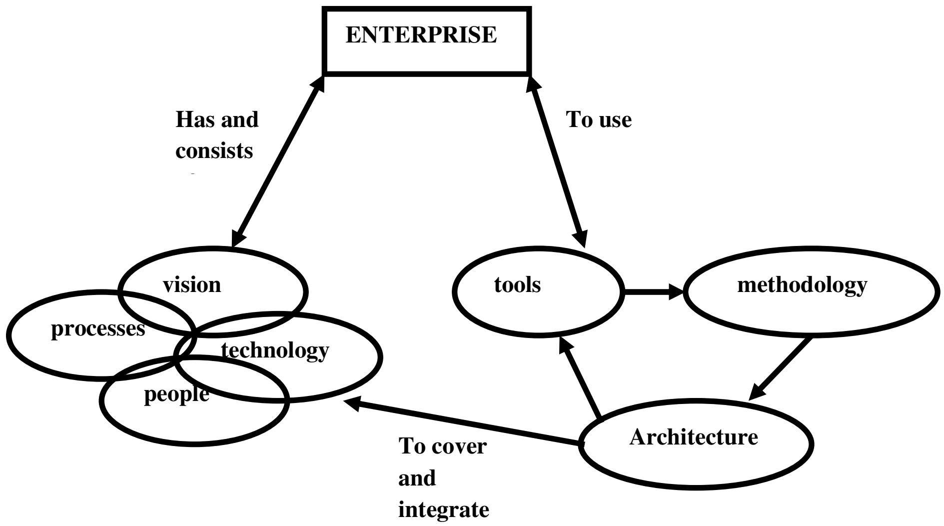


Figure 6.1 Enterprise integration framework (also see Figure 2.4).
Reference: (Ortiz, Lario, & Ros, 1999)

This thesis proposes a requirement modelling framework which is developed considering enterprise goals and objectives in relations with requirement analysis, the enterprise reference architecture that is chosen for further modelling, and finally the enterprise ontology, respectively. When enterprise vision, mission, goals; requirements analysis and representation, and enterprise ontology are combined as a whole from the enterprise modelling perspective, the conceptual framework for the requirements modelling arises as in Figure 6.2 in accordance with Figure 6.1.

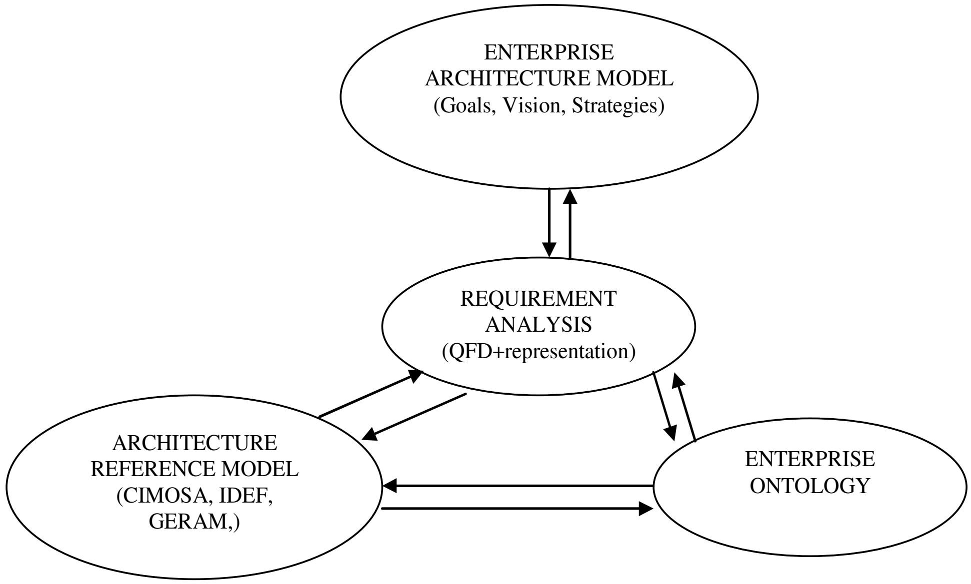


Figure 6.2. The proposed framework for modelling requirements in an enterprise.

First of all, long term goals, mission, and vision of the enterprise are clearly defined and validated by the top management through the framework in Figure 6.2. Then, processes and their objectives are described by the senior management considering the long term goals and strategies of the enterprise. The next step is to analyze user needs and expectations and the adaptation of all goals, strategies, and objectives according to these needs and expectations. This initial level can be referred to as "requirement analysis" phase. In this framework, a modified model based on Modern QFD methodology is proposed for the requirement analysis phase and explained in section 6.3 in details. The proposed QFD methodology treats the requirement analysis phase as a systematic way that manages the requirement analysis process from users' verbatim through the enterprise requirements model represented in Figure 6.2.

This thesis study synthesis enterprise modelling, organizational characteristics and definitions, and requirements analysis and modelling concepts in a single framework. The enterprise requirements framework (Figure 6.2) also emphasizes the theoretical framework of this thesis. For this purpose, enterprise modelling and reference architectures are introduced in chapter 2 and 3; then requirements analysis and QFD concepts are explained in chapter 4 and 5, respectively. All these concepts are examined to construct an integrated approach that proposes solutions for problems during requirements analysis phase of enterprise modelling. Section 6.2 addresses the motivation for an additional approach needed for requirements analysis before the particular enterprise model is constructed.

6.2 Need for Requirements Analysis in Enterprise Modelling

Inadequate and incomplete requirements are the beginning and the most important problems which canalize the designers to make mistakes during a design process, e.g., designing of a product, service, software system or an enterprise. Recent studies (chapter 4) show that requirements engineers focus on the requirement model rather than requirement analysis, and when any validation problem exists within the requirement model, this problem may probably reflect to the results / behaviour of the model. Thus, any wrong or missing statement in the requirement model would result in a gap in the final enterprise model as if a missing piece of a jigsaw puzzle.

As mentioned in chapter 2, an enterprise model is constructed to manage the processes, quality systems (ISO 9001, 14001, 18000, SA 8000,and so on), reengineering and process improvement tools such as lean production, six sigma, quality function deployment, and decision models to support the senior management, as a whole system with its suppliers, employees and customers. Any change in any component of the enterprise system would affect the interacting parts or may be the whole model. The management should easily see the effect of any change and adapt to it. The major goal of the enterprise modelling process is to adopt to the rapid change in the market and respond as fast as possible. Therefore, the system requirements, stakeholder requirements, and market requirements form the enterprise model and finally represent the infrastructure of enterprise systems.

Enterprise modelling is such a modelling process that includes all processes, systems, behaviours and the relationships among the subsystems, thus enterprise model as a whole is the largest and most detailed designing process within other system designs. The requirement analysis phase is necessary for enterprise modelling as for all design processes, but it would have a critical role on the performance of the model, when the dimensions and the detail level of the model are considered.

Analyzing and modelling enterprise requirements are the most important starting points so that the enterprise model can be used to manage the change and the relationships among stakeholders. However, enterprise requirements modelling is generally handled within the parts of enterprise reference architectures or with the traditional software engineering tools explained in chapter 4. The current studies on this field of study have some gaps between requirements model and design. This situation is exactly the same for enterprise modelling projects.

Enterprise engineers employ some tools to discover and analyze enterprise requirements, and some logical modelling and ontology structures to construct a model and test it according to the logical relationships. These requirement engineering tools are performed upon stakeholder analysis in which the users tell their expectations and problems about the system under consideration. However, these tools mainly focus on requirement representation techniques to validate user requirements without a thorough requirements discovery and analysis. A mere listing of requirements is not a systematic way for enterprise requirement analysis and architecture. Because users may not realize all they need or their requirements may not be "the requirements", a well structured requirements discovery is a prerequisite to a valid enterprise model. In this regard, approaches mentioned in chapter 4 and in this section are not thorough enough for classifying and prioritizing requirements and then converting them to design specifications.

Enterprise requirement analysis process, namely, ERA, proposes a similar framework including goal acquisition and classification, then definition of requirements with respect to the goals. This consideration is only a framework and does not suggest any technique explaining how the goals are translated (Enterprise Requirements Analysis (ERA), 2006). The suggested framework ERA in the literature can only be the starting point of requirement analysis, and for successful conversion process a quantitative method should be integrated with.

The missing points in the current status of ERA can be fulfilled by a systematic analysis model. In such a model, long term and process goals should not be only classified, but also prioritized to rank projects. Furthermore, prioritized goals should be analyzed with respect to the relationships with the short term process goals. After the final goal statements are obtained, these goals should be translated into the modelling constructs, i.e.,, functional, informational, and organizational characteristics. Therefore there exists a gap between the structured goals of the enterprise and the general aspects of enterprise modelling because of the absence of any technique that translates the goals and characteristics of the enterprise into the enterprise modelling aspects and constructs. Enterprise-QFD is proposed to fulfil this gap with its integrated structure.

The success of the requirements model depends on the success of the analysis of stakeholder/user requirements. Therefore, an enterprise engineer should also be a successful requirement engineer at the beginning of the modelling process, and the user or stakeholder analysis phase should also be handled with a more detailed and systematic approach as well as the way that is preferred in requirement modelling. In this study, a Modern framework for modelling of requirements is proposed within enterprise modelling process, and within this framework, a novel approach for requirement analysis phase is proposed called "Enterprise-QFD", which analyzes user expectations, defines their needs and deploys them into the prioritized enterprise goals, process goals, processes, and all aspects related to enterprise modelling.

6.3 Enterprise-QFD

Conventional requirements analysis and modelling methodologies have a number of weaknesses. One of the main problems with conventional approaches to requirements engineering is that each methodology seems to focus on a different aspect of development problem domain. For example some methodologies are steered in the direction of structured systems while others are concerned with object oriented systems, and some focus on job satisfaction. The technical specialism that they assume can greatly affect the way requirements are gathered with these methodologies. QFD on the other hand do not stem from any particular approach to systems development. QFD is a basic approach to collecting, documenting, analysing, prioritising and negotiating customer needs and the responses to these needs. This approach to requirements analysis is much useful than the conventional approaches. Any technical development approach can be chosen for the further steps as soon as the requirements analysis process is over.

Traditional approaches explained in chapter 4 to requirements involved the development of the requirements list, which is a very long list of customer requirements written in plain language. These lists do not scale up to convey functional and non functional requirements and are very long, complicated and unreadable and furthermore, do not represent the whole framework. In QFD customer requirements are structured hierarchically using affinity diagrams and prioritized mathematically using multicriteria decision making methods. The higher level requirements are the ones included in the matrices thus avoiding long and complicated lists of requirements. Another advantage to using affinity diagrams is that the level of abstraction at which the requirements are specified is very clear.

Use cases (in chapter 4) themselves have a downside to them. Use cases describe functional requirements from the perspective of the customer, yet they do not provide a clear separation between customer needs and the responses that a company may have to those needs. Using them alone is not enough to go deep into customer requests. This is not to say that use cases are not a good thing to have, on the contrary, they have greatly improved the requirements process making the communication between analyst/developer/customer much clearer and understandable. Use cases can also be integrated with the requirement analysis results gathered from QFD. With QFD, there is a clear separation between customer needs and technical suggestions allowing developers more creativity and space in terms of responding to these needs. Using QFD the prioritisation of the needs can now be made much more clearly. In QFD, customer requirements are explored using suggested techniques that can be integrated with QFD while with conventional approaches the classification is based on functional and non-functional requirements. QFD handles each technical component and characteristics from a customers' perspective.

QFD offers a number of tools that are not available to traditional requirements engineering methodologies. This information does not mean that QFD should replace conventional approaches; in contrast, it shows that QFD offers some tools and techniques that can be very useful especially in the requirements elicitation. QFD can become an excellent complement to the conventional approaches and that system analysts use it successfully and find it very beneficiary. One particular survey (Haag, Raja, & Schkade, 1996) has shown that using QFD in software development (SQFD in particular) yields better results than conventional software development approaches in communicating with technical people; communicating with users and meeting user requirements; communicating with managers; and finally developing consistent and complete documentation for a particular system.

Since QFD has been employed in product design and software development successfully as mentioned in chapter 5, it can also be extended for enterprise modelling to improve the requirements analysis approaches in chapter 4. QFD involves not only requirements gathering (collecting, classifying, etc.), but also the design process from the conception to marketing through some structured forms with valid information. If the enterprise model itself is considered a product, QFD can be applied in its design and creation. In this regard, there is only one study integrating QFD with one of the enterprise reference models, IDEF (Sarkis, 1993a), (Sarkis, 1993b), and (Sarkis & Liles, 1993). Yet they cover only the process model through IDEF_0 rather than the entire enterprise model, and incorporate the traditional QFD in support of IDEF only. As for the proposed methodology, Enterprise-QFD, which employs the Modern QFD, is independent of any enterprise reference architecture. The proposed methodology also modifies and improves the Modern QFD for enterprise requirement analysis and modelling.

Enterprise-QFD, developed based on third generation/Modern QFD by modifying existing parts and adding new matrices with respect to the enterprise modelling, is totally different from the QFD techniques that are developed for product/service or software design. Enterprise-QFD first collects the voice of users and analyzes and classifies them by converting them into clarified need statements, then finds out if the goals are related to these needs. This phase is called preparation phase before the quantitative analysis within Enterprise-QFD. The quantitative analysis starts with prioritization of long term enterprise goals. Enterprise-QFD employs Fuzzy-AHP technique (see chapter 5) for this purpose. Prioritized goals are handled to the first matrix where the long term goals are converted into the process goals, and the matrix chain is started. In the other phases, process goals are converted into processes, and then into modelling constructs, functional requirements, informational requirements, resource requirements, and organizational requirements, respectively. Thus, desired modelling constructs are obtained through the integrated matrices analyzing the relationships, benchmarking with the competitors and planning about the future, conflicts, technical challenges and advantages of requirements, as long as the adequate data are collected. Consequently, design targets of each requirement are determined according to the final importance values. The evaluation measures are calculated as ratio scales for each evaluation using Fuzzy-AHP, and importance values are obtained as linear distribution of the evaluation values on requirements. For each requirement, a measurement unit is defined and a target is determined at the end of each matrix. At the end of the Enterprise-QFD phases, requirement characteristics of each modelling construct is defined with the importance values.

Enterprises comprise major goals and strategies for long term process goals, and for short term plans. First of all, the major requirements should be compatible with these goals, or goals should be defined according to the requirements. Thus, the analysis phase of enterprise requirement modelling should be started with this. Enterprise–QFD starts with gathering the requirements from the stakeholders and then compares them with the predetermined long term goals. Since the enterprise is managed according to the long term goals, the analyst should check each requirement against each goal to ensure their correspondence.

After the goals are validated with the requirements, they are prioritized by a systematic approach. Enterprise-QFD uses Fuzzy-AHP for this purpose, which is explained in chapter 5. Using these priority values, and the quality matrices of QFD, these long term goals are deployed to the process goals. The quality matrix covers different parts such as relationship matrix, benchmarking matrix, technical advantage and challenge matrices and correlation matrix. According to the detailed information gathered for competitors and the mission of enterprise benchmarking, technical challenge and advantage matrices can be preferred to be used. However, the major deployment process is performed through the priorities as the input and relationship matrix. During the fulfilment of the relationship matrix, each goal is evaluated via each process goal, and the relationship degree between each couple of goal and process goal is assigned with respect to the predefined scale representing the evaluation result.

The major advantage of prioritizing long term goals and discovering the relationship between the long term goals and process goals, this process supports the enterprise engineer for verification of requirements. Thus, the engineer can find out a long term goal that is not any importance compared with the others. Furthermore, the relationship matrix indicates the degree of the relationship between long term goals and process goals, and then the engineer can reveal a goal not related with any process goals, or a process goal that is not related with any long term goal. The roof of the matrix includes correlation matrix where the process goals are evaluated in each other to handle the couples supporting each other or conflicting with each other. This part reveals the facts when any change is considered for a goal.

According to the aspects of enterprise modelling, the importance values of process goals are used as the inputs for the next phase consisting of deploying the process goals to processes and after the required calculations in this matrix, weights or importance values are obtained for the processes. Thus, deploying a characteristic to another one refers to obtaining weights of particular characteristic with respect to the evaluations, relationships, and weights of the previous modelling characteristic.

The major step to go into enterprise architecture aspects starts after finding the weights of processes. After the processes the calculations are continued on deploying the processes in the enterprise into the functional view aspects indicating each functional object and its importance weight value.

The most detailed characteristics of the enterprise are obtained in the functional aspects deployment. Since the other aspects are closely dependent on functional aspects, then the functional aspects are deployed on the other modelling aspects. The objects of information view, organizational view and resource view are then obtained with their importance values.

6.3.1 The Information about the Case Study and Company

The business application in which Enterprise-QFD approach has been implemented has been carried out in a small business company (Guven Haddecilik San.Tic.AS), which manufactures steel products in various shapes and dimensions. The customers of the company are companies manufacturing special-purpose machines, e.g., milling, lathe, drilling, and cutting. The production in the steel processing company obeys strict standards and quality specifications. Any quality problem causes customer loss.

The company management determines the short term and long term goals. They also wish to see how the goals are supported by the processes, how they match with each other; when any change is required in the firm, which of the processes and characteristics can be affected by this change.

Periodic visits have been performed to collect data. During the current status analysis phase, all processes and the Quality System Manual were examined, a form was designed to collect primary and indirect tasks, problems, needs, and the strategic road map was collected for short and long term targets and strategies.

Enterprise modelling is a new concept for Turkey and especially for SMEs (Small and Medium Enterprises). Because of this fact, the management didnot see the benefits in the beginning. However, explanations about enterprise modelling and introduction meetings have changed the attitude of the top management. The top management has recognized that the processes defined in ISO 9000 quality management system can be handled in this project. Some of the units (personnel affairs and accounting) have then been discarded from the project because they are not audited by the quality system and those are the company's own private data.

In summary, it is decided that the scope of the project is limited with the major processes structured in the quality system manual. As the properties and main purpose of enterprise modelling is introduced in chapter 2, it is a kind of project handling to manage the subsystems, processes, and the interactions among them considering the adoption to the change within the company and the environment.

The activities of the project are updated after all limitations and constraints in the company are clarified. Interviews and observations are carried out with production workers and department managers by using standard forms for questions such as critical incident analysis form and gemba visit tables (see appendix for the form and table). All qualitative data is analyzed considering the needs of users and goals of the enterprise and then they are transformed into the enterprise modelling requirements through the proposed analysis method.

Modern QFD is also modified for requirement analysis phase of enterprise modelling by changing the definitions of the columns in the CVT and MVT. The matrices are then redeveloped by reconstructing the definitions. Reordering of the matrices determining the transformations regarding enterprise reference architectures is also proposed. This new implementation of Modern QFD is called "Enterprise-QFD" because of its properties and purpose of use. In section 6.3.2, details of its implementation in the company are presented with the systematic steps of Enterprise-QFD developed based on Modern QFD.

6.3.2 Enterprise-QFD Steps

In this section, Enterprise-QFD phases are introduced based on the data gathered from the case in section 6.3.1. Through Modern QFD, requirements analysis is performed with a few modifications in the preparation phase (steps 1 to 7 in the following). In the last phase, the content, name and structure of the QFD matrices are developed from scratch for enterprise modelling aspects in Enterprise-QFD. The Modern QFD based calculations are then performed in these matrices. The QFD matrix calculations are improved to avoid its sceptical mathematical scale, which is changed from ordinal scale to ratio scale in Modern QFD. Furthermore, the ratio scale statements are generated by using Fuzzy-AHP in the Enterprise-QFD matrices.

Enterprise-QFD follows the same phases of Modern QFD at the beginning of the analysis by some modifications. Modern QFD and Enterprise-QFD handle each requirement analysis study as a project, and starts the steps by defining the goals of the project. In the second step, user segments are determined with the characteristics and the management decides which segment(s) will be considered in the modelling. In the third step, the verbatim is collected from the target segment(s). These steps are followed by clarification of the verbatim, analysis of the verbatim with respect to the enterprise goals, the prioritization of the goals, and the transformation of the goals into modelling issues through the evaluations and sequential matrix calculations.

Enterprise-QFD phases are introduced below, in accordance with the information gathered from the company mentioned in section 6.1.

1) Project Goals:

In the first phase, the management of the company defines needs for the enterprise model. The project goals should not be confused with the goals of the enterprise. Some of the most significant enterprise modelling project goals can be defined initially: “manage enterprise integration”, “construct controllable processes”, “adopt standardization", and "improve worker/customer satisfaction". These goals should be measurable and/or visual so that the improvement can be traced and managed. The goals of the company are defined and represented in Table 6.1.

Table 6.1 Goals of enterprise modelling project

Definition	**How to measure?**	**Time frame**	**Who will judge the result?**
To trace the integrated relationship among the divisions and processes.	No. of problems faced	6 months	Top management, quality system manager, production manager
To manage the change and revisions efficiently	Time required for revisions	6 months	Top management, quality system manager, production manager
To show the processes to the customers and suppliers throughout a model	The time for introducing the firm and processes.	6 months	Top management, quality system manager, production manager

2) Identify User Segments:

This phase defines the users for whom the enterprise model is used. Customers, management, employees, suppliers, and all other stakeholders are the potential segmentations in an enterprise. Enterprise-QFD does not only list the stakeholder names but also analyzes their relationships with the enterprise using a standard table. This table clarifies the customer segment by asking the questions such as "who is the customer?", "what is the enterprise model used for?", "When is the model used?", "Why and how is the model used?". From the viewpoint of enterprise modelling, relationships between customers/users with the enterprise should be identified (Table 6.2).

Company sizes may affect the segmentation, further segmentation may be added especially for the customers who provide different value for the company.

3) Go to gemba:

Gemba is the place where the product becomes a value from the viewpoint of the customer (QFDI, 2002). Therefore, it is the place where customers use the product, or the place where the product is processed (for internal customers). For Enterprise-

QFD, gemba is the enterprise and its environment, i.e., all staff in the enterprise and customers of the enterprise.

Table 6.2 Customer segments table

Who	What	When	Where	Why	How
Personnel and department managers	The whole or partial model of the enterprise	When information or decision is required that covers one or more processes	In the office or shop floor	To track change in process(es), effect of the decision or information.	By examining the enterprise model on the paper to computer screen
Top management	The whole or partial model of the enterprise	Any decision about management of the enterprise	Meetings, project or report presentations	To make a follow-up methodology for all business over the long term goals, strategies, and their potential effects on processes	By comparing the long term goals with process goals, outputs defined in the enterprise model
Suppliers	For the related processes where the items are supplied for.	When a corporation in improvements or contracts are handled with a supplier	In evaluation meetings with the company, or during the inspections in shop floor	To understand the business and processes of the company and to plan its own business considering these information	Company may present a sub model related to the supplier's material. Supplier can analyze the business over the model.
Customers	For improved quality of products and satisfactory relationships	During customers' visits or audits within the company	In evaluation meetings with the company, or during the inspections in shop floor	To understand the processes and their interactions, quality standards, products and other related business in a short time	By examining the enterprise model on the paper or computer screen
The requirements model of the corresponding company has been constructed with respect to the evaluations of personnel and senior management in the company.					

The Gemba, which covers the user segment(s), is observed to discover the user needs. They are arranged in a structured table form called the “gemba visit table” to systematize the observation process. Each column is independent from each other and used to clarify the voice of the user observed. A gemba visit table includes a part introducing the customer with his/her detailed contact information. It includes another part constructed as columns to record the verbatim of the corresponding customer expanded with observations notes, provided documents related with a specific process steps, and finally clarified items retrieved from verbatim. A blank gemba visit table is given in Appendix1 . A gemba visit table is frequently used in the Enterprise-QFD project during observing the enterprise environment, and as a sample, Table 6.3 shows the functionality of this table within the requirement analysis.

Table 6.3. Gemba visit table

Interviewee: Quality System Manager **Contact Info:** **Interviewer:**			**Date:** 12.09.2006 **Place :** Office		
Interviewee Characteristics: High tempo working conditions, multiple tasks and responsibilities, observations and evaluations in shop floor.					
Environment: Flexible and relax office environment.					
Process Step	**Observations**	**Verbatim**	**Documents**	**Notes**	**Clarified items**
Quality Control plan and reports	- Material certificates. - Irregular production plans - Reworks but no scrap - Frequent controls to avoid the defects	- Initial quality control - Final controls - Process controls - Material analysis for the imported parts.	- Material certificates - Analysis Reports	Rarely faced communication problems with operators	Material certificates Material analysis Process control Production follow-up

Table 6.3 is a part of gemba visit table showing a partial interview with the quality system manager of the company about quality items and characteristics. A completed gemba visit table can be examined in appendices.

4) Customer Process Model:

A customer process model is constructed through data flow diagrams, process flows, and other tools to determine common level of customer understanding. These tools are used to validate the model, i.e., they warn the analysts when they are out of the gemba. Users cannot always articulate what they need, which makes it difficult to perform the gemba visits. In this case, some user stories can be created to discover what the users really mean. All the statements noted are then translated into clarified items to be validated by the users. The purpose of this study is to analyze the requirements of an SME to model its processes. Therefore, the corresponding user or customer process model would be the processes permitted to be analyzed. In this SME, the analysis of requirements is only permitted for the predefined processes belong to its ISO 9000 represented in Figure 6.3. Solid lines indicate the direct and close relationship while dashed lines indicate indirect and weak relationships.

5) Customer Voice Table (CVT):

CVT arranges the requirement statements by decomposing them into benefits, needs, and product features gathered directly from the gemba visit table or from complaint reports, warranty data, and sales reports. CVT is also essential for requirement analysis to see which statement means what, and how they are related to each other. Verbatim of users are converted into clarified items according to the statements gathered from gemba visits. The related columns are then matched to construct some paths through the table. CVT represents the users' verbatim rather than designers' ideas. Table 6.4 shows a partial CVT table. The columns are filled in independently of each other. The paths are to determine the related columns. All possible paths are formed following the related verbatim from left to right and right to left to reach the related need.

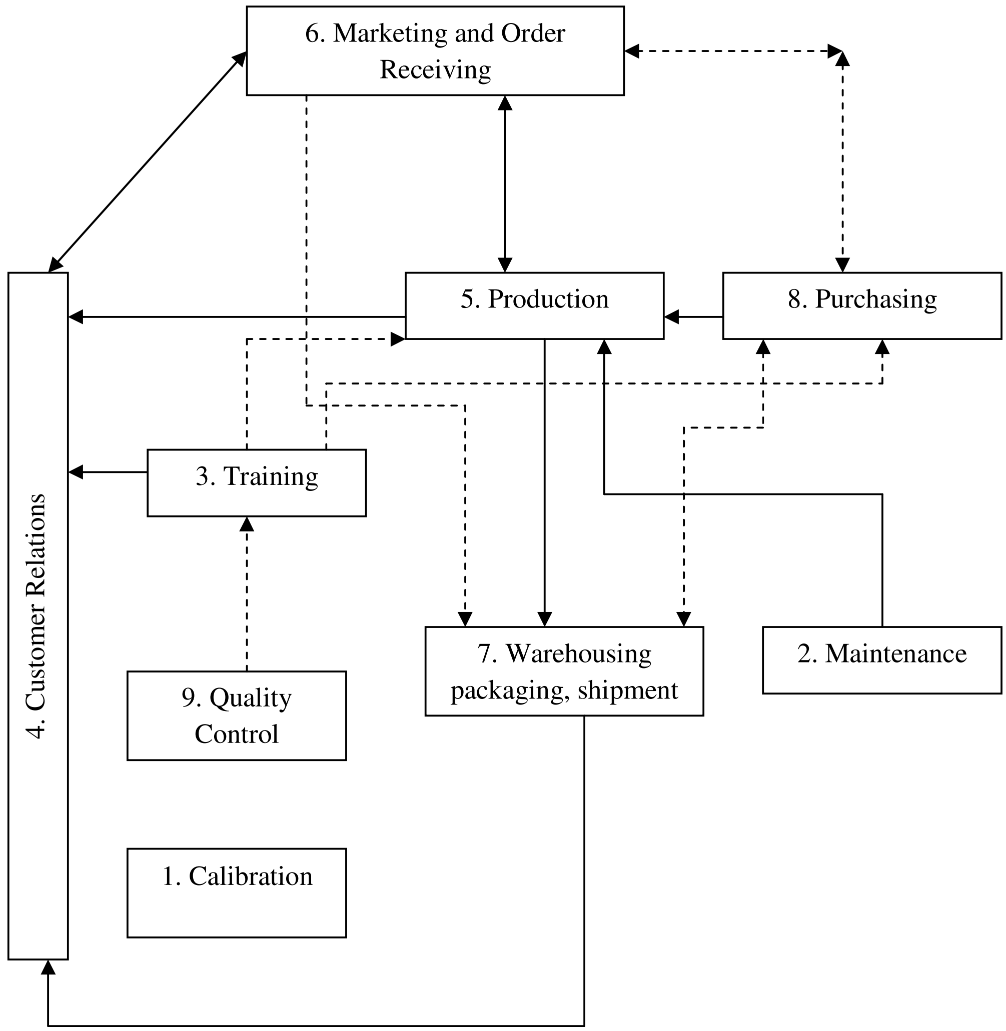


Figure 6.3 Processes of the SME company in ISO 9000 system and their interactions.

Table 6.4 Customer value table

Customer/User					Solution			Organization
Segment	**Characteristics**	**Situations**	**Problems**	**Needs**	**Characteristics and Capabilities**	**Actions**	**Technology**	
				Being a brand	Consultancy support	Attending the fairs for promotion		Marketing
Top management		The current situation of export is not adequate		Minimum number of customer complaints	Activities for increased export	Planned customer visits	CDs promoting the company	
			Unplanned relationships with the customers	Improved customer loyalty	Brochures and catalogues for promotion		E-commerce	
		Communication with the firms consulting about import and export	Deficiencies in promoting	On time delivery of products	Regarding the customer satisfaction	Efficient shipment plan for on time delivery	Web page	
		Communication with the suppliers	Material problems		Newly reengineered marketing department	Advertising in the magazines of the sector		Purchasing and Warehouse
				More sales volume	Qualified sales staff		Technological improvements on quality of the materials	
Company staff and department managers				Production with quality materials	Material analysis and certifications	Detailed maintenance plan		Quality Assurance
			Machine breakdown	Minimum machine damage in the production area	Qualified maintenance staff	Definition of the required control points		
			Manufacturing failures	Reachable maintenance equipments whenever needed	Daily/weekly production follow-up reports	Qualified worker employment	Improved machines, equipments, and tools	Production
			Old machines	Manufacturing with minimum failure		Employment of product engineer	Business management systems (i.e. ERP)	

Table 6.4 Customer Value Table (cont.)

Customer/User					Solution			Organization
Segment	Characteristics	Situations	Problems	Needs	Characteristics and Capabilities	Actions	Technology	
			Incapable of realizing planned production	To display the production realization rates and productivity	Software support for production planning and control		Business management systems (i.e. ERP)	
				To follow and analyze the performance				
			Work accidents	Minimum staff damage in the production area		Precautions for work accidents		Production
				Awareness of the staff about the improvements in their jobs	Research habits for each department	Motivation activities for the production staff		
Company staff and department managers				System and documentation success		Research for appropriate process control methods		Quality Assurance
	Multifunctional workers employing several tasks of different jobs	Adaptation of the staff to the quality system	Qualification problems of manufacturing workers	High employee satisfaction	More isolated organizational structure	Planned training activities		Marketing
Top management				Adaptation of personnel to the quality system	Weekly evaluation with management			Purchasing and Warehouse
			High turnover for workers	Minimum intersections among the tasks of the departments	Predefined goals and action plans for the departments			
			Structural disorder in organization		Improved quality management system	Revisions within the system		

In a customer value table, e.g. Table 6.4, the verbatim of users/customers including the top management are placed with respect to the characteristics of the verbatim. Any verbatim may be just a clarified need, represent a problem or a solution for a potential problem, or a characteristic of the system. Even if the users may consider a technological issue or a required action, hopefully the customer value table provides the necessary platform to put all these issues in one table to divide the verbatim into clarified categories and then to translate all categories in terms of the needs by matching them.

6) The structure of the needs (affinity diagram), hierarchy diagram, and priorities:

Enterprises may have numerous requirements. The top management cannot consider all, but selects the most important ones evaluated in the next phases. Affinity and hierarchy diagrams help to classify the clarified requirements in mutually exclusive groups and construct the hierarchy structure within these groups. Because requirements are not equally significant in a hierarchy level, some importance levels should be calculated after the classification through a multi-criteria decision making tool, e.g. Analytical Hierarchy Process (AHP), ANP, and Fuzzy-AHP as mentioned in chapter 5. The proposed approach employs Fuzzy-AHP for prioritizing the goals as the starting point of the enterprise requirements in Enterprise-QFD.

7) Maximum Value Table (MVT):

This step is to share the information gathered from the customer and arranged in the CVT with the model designers. MVT is constructed to classify what functional requirements exist, and what kind of features should be specified (functions, tasks, processes, and entities) from the designer's perspective. It is the road map of designers. MVT was introduced by Modern QFD, and is not available in the classical QFD.

This study modifies some columns of the original MVT for enterprise modelling. In the original MVT, the columns are generated especially for product/service design characteristics, and cannot be used in enterprise modelling. Some columns are deleted, and another column called "enterprise goals" is added to the MVT so that the transformation of the enterprise characteristics into enterprise modelling aspects is started with the goals of the enterprise. Before the enterprise model is translated into a reference model, MVT for the vital needs should be constructed for the enterprise engineer to comprehend all interactions in the enterprise. Table 6.5 shows a partial MVT of Enterprise-QFD. It is similar to the customer voice table. Yet the customer voice table is prepared according to the voice of customers whereas the MVT is prepared based on the chosen design characteristics of the model by the designers. The columns are filled in independently from each other. Some paths are then drawn to determine the related columns.

MVT is very important before further analysis, because the goals are validated by the user needs. Furthermore, this is a check point for an enterprise engineer if there is any missing matching between any need and enterprise goal. Thus, the enterprise engineer and management can see if there is a need which is not considered in goal statements, or a goal statement which is not related to any of the needs.

All requirements should be met if and only if they are represented within the enterprise goals. MVT guarantees that all the goals cover the needs. If the enterprise goals were not predefined, then the classified needs would be prioritized first, and the needs would be converted into enterprise goals through an additional QFD matrix.

Table 6.5 Maximum value table

Customer/User					**Solution**			**Enterprise Goals**
Segment	**Characteristics**	**Situations**	**Problems**	**Needs**	**Characteristics and Capabilities**	**Actions**	**Technology**	**(Long-term)**
				Being a brand	Capable of having external support for promotion and export	Attend the industrial fairs	CRM	To have a competency advantage (0.19)
Top management		The current situation of export is not adequate		Minimum number of customer complaints		Plan and make customer visits	Company CD	To be a brand (0.0001)
			Unplanned relationships with the customers	Improved customer loyalty	Being customer oriented	Prepare a CD to promote the products and company	E-business	Increase on export volume (0.14)
		Communication with the firms consulting about import and export	Deficiencies in promoting	On time delivery of products		Make a shipment and inventory plan with suppliers		
		Communication with the suppliers	Material problems		Experienced marketing staff	Provide advertising in the magazines of the sector		
				More sales volume	Qualified sales staff	Make agreements to design a web page	Analysis laboratory for material analysis	
Company staff and department managers				Production with quality materials	Requiring material analysis reports from suppliers	Employ qualified worker		Increased production capacity (0.08)
			Machine breakdown	Minimum machine damage in the production area		Define control points for each process		
			Manufacturing failures	Reachable maintenance equipments whenever needed	Qualified maintenance staff	Make a detailed maintenance plan	Machines and equipments with new technology	Improvement of product quality (0.59)
			Old machines	Manufacturing with minimum failure	Reachable Daily/weekly production follow-up reports	Provide a product engineer		

Table 6.5 Maximum value table (cont.)

Customer/User					Solution			Enterprise Goals
Segment	**Characteristics**	**Situations**	**Problems**	**Needs**	**Characteristics and Capabilities**	**Actions**	**Technology**	**(Long-term)**
			Incapable of realizing planned production	To display the production realization rates and productivity	Capable of following the production status on screen display		Business management systems (i.e. ERP)	To have a competency advantage (0.19)
				To follow and analyze the performance				
			Work accidents	Minimum staff damage in the production area	Trained workers about occupational safety	Provide precautions for work accidents		Increased production capacity (0.08)
				Awareness of the staff about the improvements in their jobs	Production staff willing to improve their experience and knowledge	Motivation activities for the production staff		
Company staff and department managers				System and documentation success		Research for appropriate process control methods		
	Multifunctional workers employing several tasks of different jobs	Adaptation of the staff to the quality system	Qualification problems of manufacturing workers	High employee satisfaction	Uncomplicated organizational structure	Plan training with respect to the requirements		Improvement of product quality (0.59)
Top management				Adaptation of personnel to the quality system	Evaluation meetings with shop floor staff			
			High turnover for workers	Minimum intersections among the tasks of the departments	Traceable process goals for each departments	Make revisions within the system		
			Structural disorder in organization		Improved quality management system			

Recall that enterprise modelling differs from traditional product, service, or software design process in some aspects. Enterprise modelling is a generic approach to show how the things go on in the company by considering the processes and the subsystems working on these processes. Needs of the processes and decisions of top management are considered with higher priority than the bottom level employees. In traditional enterprises the business is handled through the decisions of owners or top management, and their goals are the explanatory targets that determine the future work of the enterprise. Despite the fact that Modern QFD analyzes all user needs with equal importance, even for the workers at the bottom level, the design points start with the goals of enterprises defined by top management. Therefore, in MVT (Table 6.5) after CVT (Table 6.4), customer needs and other design issues are matched with predetermined ones set by managers. But Modern QFD tables still provide a platform where the needs of all people in the company can be tested for matching.

Both CVT and MVT explain capabilities/characteristics, and actions of the company and needs behind them. CVT represents the user verbatim whereas MVT represents the design decisions of the modelling team such as enterprise goals. In standard MVT and CVT, there exist design fields to show the design issues such as functional and technological components of a product or service, but in Enterprise-QFD "enterprise goals" column exists additionally. This is because of the properties of enterprise modelling, and modelling constructs defined in this process. Modelling constructs relies on functional, informational, resource, and organizational views of enterprise, and design is dependent on these constructs. Enterprise-QFD handles the transformation process by using QFD matrices during the design phase.

Enterprise-QFD also considers the enterprise goals during the design process, and then the goals are the starting point of the transformation. Regarding this consideration, MVT is redesigned to match the needs, characteristics, actions, and other fields about the enterprise with the enterprise goals to test whether the goals meet the needs or not before transferring the goals into modelling characteristics through QFD matrices. The modified MVT in Enterprise-QFD ensures that, , the requirements of an enterprise can be met if and only if these goals are defined as enterprise goals. MVT carries out this mission through a goal column that is added in the last step of the MVT. Thus, each road map in MVT shows the path from a requirement to a particular enterprise goal.

After the goals are clarified in MVT, they are prioritized by using the AHP technique. It was seen that one of the goals did not have any significance and was not highly related to the process goals. Hence, it was discarded. The priority numbers are the first input of the matrix calculations where the long term goals are evaluated according to the relationship with process goals. This is the first matrix on the way to define the enterprise modelling characteristics. This matrix is followed by the others which are redeveloped according to the enterprise modelling constructs where the processes are decomposed into functional, informational, resource, and organizational characteristics representing all relationships in each phase with weighting numbers. The next phase explains how these matrices are constructed and calculated.

6.3.3 Matrices for Translation of Goals into Requirement Characteristics

QFD employs matrices for detailed requirements analysis that translates quality into design features. First, the enterprise is analyzed according to its long term goals, namely enterprise goals. In this regard, the needs of the predefined users/customers are combined to find out whether they are matched with any of the enterprise goal This concern is handled in the special tools of Modern QFD for verbatim analysis; Customer Voice Table (CVT), and Maximum Value Table (MVT), and the original tools are changed to perform the required analysis of enterprise goals by cancelling the product/service –purpose columns and adding goals columns. MVT is the most important phase before the matrix calculations where the goals are validated with user needs.

All requirements should be met if and only if they are represented within the enterprise goals. MVT guarantees that all the goals cover the needs. If the enterprise goals were not predefined, then the classified needs would be prioritized first, and the needs would be converted into enterprise goals through an additional QFD matrix.

After the fitness of the user needs with the enterprise goals was ensured, some serial mathematical analysis and evaluation phases are performed based on the modified and reconstructed QFD matrices. The necessary calculations are performed according to the Modern QFD. The study also modifies these houses and introduces new ones in the scope of Enterprise-QFD. The original QFD matrices convert customer requirements into product characteristics, product and production planning, and so on. Since the concern of this dissertation is to propose a methodology for the requirement analysis of enterprise modelling, the houses and its sequential structure are developed according to the proposed framework in Figure 6.2, and the aspects of enterprise modelling in chapter 2 and chapter 3. Figure 6.4 shows the proposed structure that would be used to translate the priority values of each column of each step to the reference model.

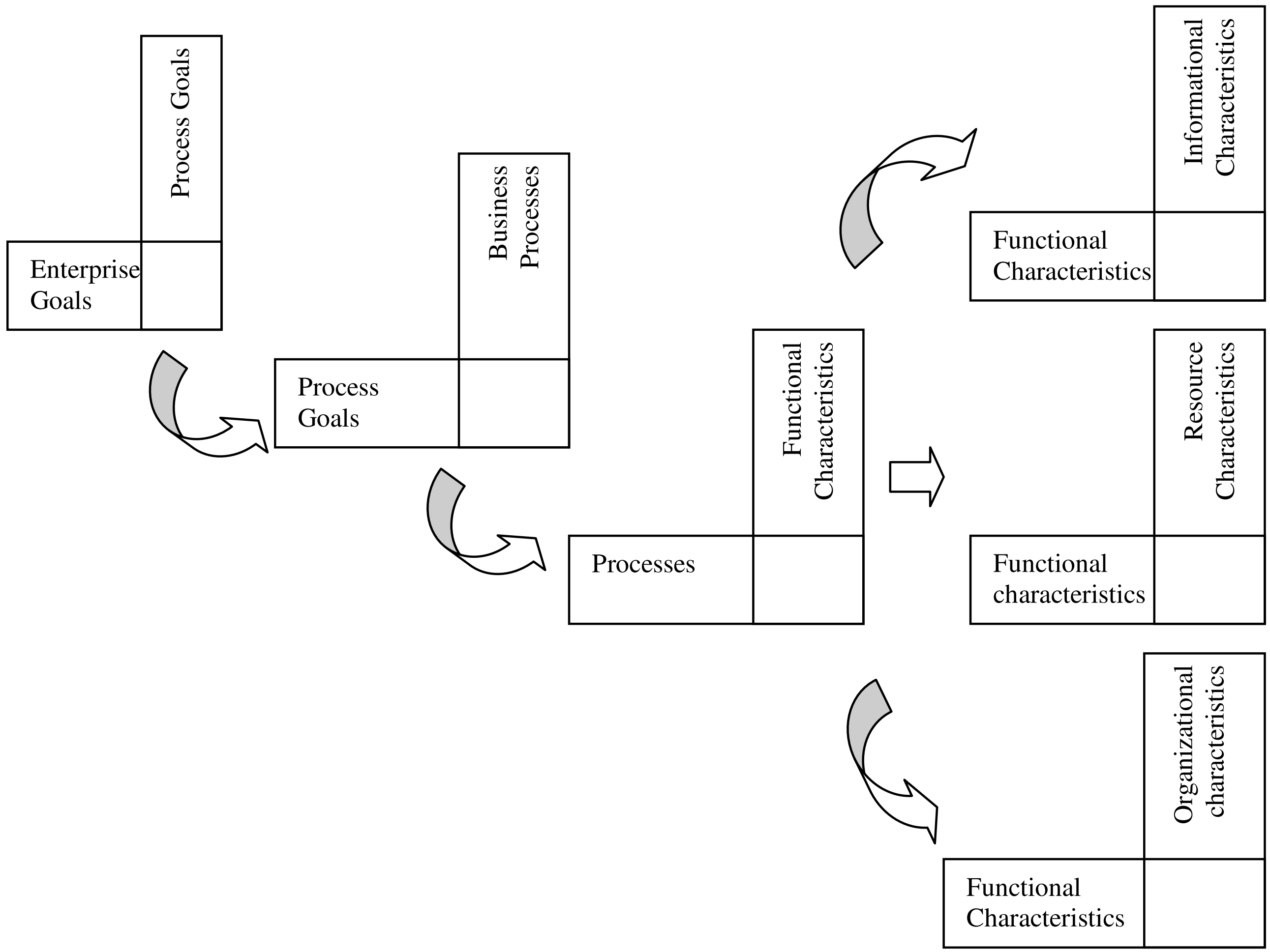


Figure 6.4 The Enterprise-QFD structure for requirements analysis.

This matrix sequence is introduced by Enterprise-QFD, and has novel contents and definitions when the following structural differences are considered:

1. Goal Deployment (Enterprise Goals to Process Goals)
2. Process Deployment (Process Goals to Business Processes)
3. Functional Deployment (Business Processes to Functional Requirements, e.g. domain processes, activities, etc.)
4. Functional Requirements to
    i. Information requirements (objects, events)
    ii. Resource requirements (definition, capacity)
    iii. Organizational requirements (structure, tasks)

The matrix in Figure 6.4 is built for only requirement definition phases. This structure has three levels in application, where the first two levels include one matrix for each whereas the third level consists of three matrices. In addition to this structure, the forth level can also be constructed to convert the requirements into a reference architecture model according to the inputs gathered from the other matrices.

This sequence and content of matrices are improved and redesigned for enterprise modelling. Requirement analysis based on these matrices is the major contribution of this thesis study. Integration of these conversion matrices with respect to the aspects of enterprise modelling is a novel approach and the calculations are not based on traditional QFD developed in 90s, but on third generation (Modern) QFD, which is frequently implemented by QFD experts in recent years.

All relationships, as well as their significance rates, and priorities, conflicts (roofs of the houses) and benchmarking issues can be considered in an integrated manner through the series of the houses (matrices). Since QFD marices are built on subjective evaluations, a scale is needed to mitigate them. According to the Modern QFD, a ratio scale is developed for each part of the QFD matrix. This study employs Fuzzy-AHP explained in chapter 5 to determine the priorities and the evaluation scale. The first step in the construction of the matrices is to develop a common evaluation scale for different parts of QFD through pairwise comparisons with the Fuzzy-AHP, shown in Tables 6.6 to 6.9. The goals are also prioritized with the Fuzzy-AHP.

Table 6.6 Scale for relationship matrix

Definition	**VALUE**
Very strong relationship	0.51888
Strong relationship	0.33389
Weak relationship	0.14724
No relationship	0.00000

Table 6.7 Scale for benchmarking and planning matrix

Definition	**VALUE**
Very high	0.54247
High	0.33996
Medium	0.11757
None	0.00000

Table 6.8 Scale for technical advantage

Definition	**VALUE**
Very big	0.51888
Big	0.33389
Little	0.14724
None	0.00000

Table 6.9 Scale for technical challenge

Definition	**VALUE**
Major	0.58193
Minor	0.41807
None	0.00000

The goals are also prioritized with the Fuzzy-AHP. Because QFD has a tailored analysis process and does not prescribe any scale, the relationships in the matrices were performed by a scale developed through the Fuzzy-AHP's pairwise comparison process to acquire user considerations to be used in the evaluations.

Each QFD matrix is evaluated according to the relationships and benchmarking issues. The following operations are then performed in each matrix. Enterprise-QFD performs sequential matrices where the results (weights of the characteristics given in columns) of any matrix is the inputs (rows in the first two columns) of the next matrix. In the evaluations, QFD performs the following operations:

1) **Priority of the inputs** (part 7 in Figure 5.2): if it is the first matrix, it is obtained through AHP/Fuzzy-AHP /ANP. Otherwise, the final normalized weights of the columns of the previous matrix are taken as the priorities of the next one. Construction of a network and calculation of complicated matrix in ANP is so difficult and inefficient that QFD practitioners do not prefer to use this methodology. The prioritization process should not be longer than the requirements analysis and representation for an efficient design process, thus, customer needs are tried to be defined by independent statements and prioritized using AHP in practice.

2) The parts of the matrix (the parts in Figure 5.2) can also be weighted through the same techniques.

The integrated matrix in Figure 5.2 consists of seven sub-matrices, namely, customer requirements list (1), technical characteristics (3), importance levels (7), relationship matrix (4), benchmarking and planning (2), technical weights and design targets (6), and correlation matrix (5). After a detailed preparation phase including project goals, customer segmentation, customer voice analysis and clarifying the needs (see section 6.2), the requirement list (1) is defined as the first part. Then, the needs are grouped into categories and the hierarchy of these categories is determined. The customers evaluate the needs and assign values using the procedure of Fuzzy-AHP (part 7). These weights constitute the importance values part in the matrix. The first two parts are the inputs of this matrix and when they are ready the evaluation phases are carried out.

First of all, the measurement scale is constructed for each different evaluation. The verbal statements are defined and then weighted by each customer by using AHP or related methods, and therefore ratio scale numbers are obtained for each type of evaluation (see section 6.2). After the technical characteristics are defined in the columns (part 3), each of them is evaluated by each customer requirement in the rows, and the relationship values (part 4) are assigned with respect to the evaluation scale. In the next phase, each requirement is handled to be compared with the competitor(s), and planning values about improvements are defined (part 2) according to the corresponding ratio numbers of verbal statements. If enterprise engineers need additional parts such as technical advantages, and challenges, then these parts may be added and further evaluations may be performed about them. The calculations are handled according to the linear distribution of ratio scale values. The final numbers in the matrix present the importance values of each technical characteristic (part 6) with respect to the customer preferences and competitors status in the market. The roof of the matrix (part 5) is used to show whether there is a contradiction between each couple of technical characteristics from the viewpoint of design process. The QFD practitioner collects the importance values and combines them with the information from the roof, and then defines the value of each design issue about technical characteristics. The mathematical calculations and formal statements of the matrix can be found in section 6.5 in details. Any matrix may be added after another one, if the output of one matrix can be used as the input in the other one for further modelling such as product planning phase after design phase. The content of the matrices are redesigned according to the issue to be designed and the level of details, but the same calculations are performed.

3) The adjusted importance weight (using part 1, 2 and 7 in Figure 5.2) for each row is then calculated through Equation 5.1 and 5.2.

4) Relationship values are assigned to part 4 in Figure 5.2 between the rows and columns using the scale in Table 6.6.

5) Absolute importance weight for each column is calculated by Equation 5.3.

6) The adjusted weights of the columns are calculated based on the absolute weights. The evaluations of part 6 in Figure 5.2 including the weights of functional characteristics, technical challenge and technical advantage, and their evaluation values in Table 6.8 and 6.9 are then calculated by Equation 5.4.

7) The roof of the house (part 5 in Figure 5.2) is considered only when the design targets are determined. Otherwise, it does not take part in the calculations.

8) The design targets are defined. The numerical values obtained in Equation 5.4 are transferred as priority values (P_i) for the next matrix.

From the project goals to the maximum value table, all activities get related by discovering the user requirements. The QFD matrices are then utilized from the goals through the model components for the detailed analysis. In the first matrix of Enterprise-QFD, Table 6.10, the enterprise goals are prioritized and converted into process goals including medium to short period goals. Many companies draw a "strategic road map" to manage them. The standard strategic road map can be used at the definition phase of the goals. The matrix then figures out the relationships between the enterprise goals and process goals considering the status of the competitor, improvement targets, importance of the enterprise goals, and conflicts from the roof. The mathematical integration of all these measures determines the functional characteristics weights. Then, technical challenge and technical advantage issues are evaluated for the final absolute weights according to the competitor's scores.

In the last step, the design targets on the process goals can be defined. One should not forget that any QFD house is an integrated matrix including many parts (parts 1 to 7 in Figure 5.2) and these parts can work independently according to the detail level of the analysis. The main purpose of the QFD matrix is to figure out the relationships to convert the concepts from the rows into the design issues in the columns. The design targets are then decided according to the relationships. Thus, the compulsory part of the matrix is the relationship part (part 4 in Figure 5.2) used for the conversion. In the evolution of the QFD matrix, the other parts (competitive improvement, technical challenge, technical advantage, etc.) have been added for a more detailed analysis. If any part of the matrix cannot be fulfilled; except the importance values, relationship values and design targets, then the analyst can cancel this part and handle the operations on the other parts. Therefore, if any problem is faced during data collection or evaluation in the detailed level, these parts can be cancelled without stopping the analysis process. As seen in the matrices from Table 6.10 to Table 6.15, the roofs of the houses cannot be utilized because of the limitations of the data in the company where the sample case was retrieved.

The results of the analysis through the matrices of Enterprise-QFD are presented respectively in Tables 6.10 through 6.15. These matrices indicate the relationships between the enterprise goals and process goals, process goals and processes, and finally processes and enterprise modelling issues such as functional, organizational, resource, and informational. This sequence of matrices supports the enterprise engineer during the management of enterprise modelling process considering the enterprise and process goals.

The first matrix is fulfilled to convert the enterprise goals into process goals. The calculations start with finding the importance level of each enterprise goal by pairwise comparisons in Fuzzy-AHP. If some goals are evaluated to be zero, then this means that these goals are not significant and can be discarded from the goal list. In the case study, the company managers had already defined the enterprise goals for their strategic road map given in the last column of MVT in Table 6.5. However, it was then seen that the goal defined as “to be a brand” had nearly zero importance level. Besides, weak relationships were observed between this goal and each other process goal in Table 6.10 (the shaded area). Thus, this goal and its related row is discarded from the goal list and not considered for the calculations in Table 6.10. On the other hand, any characteristic in a row with high importance level might also be weakly related to other characteristics in the matrix. All these contradicting situations cause the analyst to go back and check the goal definitions at the beginning. This property can also be utilized in the other Enterprise-QFD matrices.

Table 6.10 Conversion of enterprise goals into process goals

	Priorities		Employment of new marketing and sales staff	Reengineering of marketing division	Determining the requirements about the products	Determining the competitors and performing analysis	Consultancy support about creating a brand	Determining the advantages and disadvantages of the company with respect to the competitors	Attending to the sectoral fairs	Advertisments in the sectoral magazines	Training support for manufacturing staff	Employment of product engineer	Improvment of machines and equipments	Testing the quality of materials	Efficient calibration	2 shifts a day	Employment of new staff for manufacturing	Research for new technologies	Performing market research	Customer visits	Communication with new customers	Defining the customer needs	Planning Table						
		0,55																					Competitive Improvement					**0,45**	
	Local	Global																					Customer/user evaluation of current status of the enterprise	Customer/user evaluation of current status of the competittor	Positioning plan	Competitive Improvement	Local Improvement	Global Priority	Adjusted weight
Competency advantage	0,19	0,10	0,33	0,33	0,33	0,52	0,52	0,52	0,33	0,33	0,33	0,33	0,33	0,15	0,15	0,15	0,00	0,33	0,52	0,52	0,52	0,52	0,12	0,34	0,34	0,15	0,19	0,09	0,19
Production quality improvement	0,59	0,32	0,00	0,00	0,00	0,15	0,00	0,15	0,15	0,00	0,52	0,52	0,33	0,52	0,33	0,15	0,15	0,52	0,15	0,15	0,00	0,00	0,34	0,34	0,54	0,33	0,43	0,19	0,52
Increased production capacity	0,08	0,04	0,15	0,33	0,52	0,33	0,00	0,15	0,33	0,15	0,15	0,33	0,52	0,15	0,33	0,52	0,52	0,33	0,33	0,15	0,15	0,15	0,12	0,12	0,34	0,15	0,19	0,09	0,13
Increased export	0,14	0,08	0,52	0,52	0,52	0,33	0,15	0,15	0,52	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15	0,33	0,33	0,33	0,15	0,12	0,34	0,34	0,15	0,19	0,09	0,16
Absolute weight			0,17	0,19	0,22	0,27	0,12	0,22	0,27	0,11	0,35	0,37	0,30	0,32	0,24	0,17	0,14	0,40	0,27	0,25	0,17	0,14	**4,70**			**0,78**	**1,00**	**0,45**	**1,00**
Functional requirement	local		0,04	0,04	0,05	0,06	0,03	0,05	0,06	0,02	0,07	0,08	0,06	0,07	0,05	0,04	0,03	0,08	0,06	0,05	0,04	0,03	**1,00**						
0,66	global		0,02	0,03	0,03	0,04	0,02	0,03	0,04	0,02	0,05	0,05	0,04	0,04	0,03	0,02	0,02	0,06	0,04	0,04	0,02	0,02	**0,66**						
Current Status			0,00	no	no	no	no	no	1,00	3,00	24,00	no	0,00	0,34	0,52	1,00	3,00	0,00	0,15	3,00	0,15	0,34							
Competitor A			yes	yes	-	yes	no	-	1,00	2,00	24,00	yes	0,00	0,34	0,34	1,00	0,00	0,15	0,34	5,00	0,34	0,34							
Design Target			1,00	yes	yes	yes	no	yes	3,00	3,00	30,00	yes	0,15	0,52	0,52	2,00	3,00	0,15	0,34	6,00	0,34	0,34							
Measurement			no. of staff	yes/no	yes/no	yes/no	yes/no	yes/no	no. of att.	no.of ad.s	total time (hr)	yes/no	linguistic	linguistic	linguistic	no.of shifts	no. of staff	linguistic	linguistic	no.of visits /year	linguistic	linguistic							
Technical advantage			0,15	0,00	0,00	0,15	0,00	0,15	0,15	0,33	0,15	0,15	0,15	0,15	0,33	0,00	0,00	0,15	0,15	0,33	0,33	0,15	**2,96**						
			0,05	0,00	0,00	0,05	0,00	0,05	0,05	0,11	0,05	0,05	0,05	0,05	0,11	0,00	0,00	0,05	0,05	0,11	0,11	0,05	**1,00**						
		0,29	0,01	0,00	0,00	0,01	0,00	0,01	0,01	0,03	0,01	0,01	0,01	0,01	0,03	0,00	0,00	0,01	0,01	0,03	0,03	0,01	**0,29**						
Technical challenge			0,06	0,08	0,08	0,08	0,08	0,08	0,08	0,06	0,00	0,00	0,06	0,00	0,00	0,00	0,00	0,08	0,08	0,06	0,08	0,06	**1,00**						
		0,05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	**0,05**						
Adjusted weight			0,04	0,03	0,03	0,06	0,02	0,05	0,06	0,05	0,06	0,07	0,06	0,06	0,07	0,02	0,02	0,07	0,06	0,07	0,06	0,04	**1,00**						

Table 6.11 Conversion of process goals into processes

	Priorities	1										
	Local	Global	Calibration	Maintenance	Training	Customer Relations	Production	Marketing and order receiving	Warehousing, packaging, shipment	Purchasing	Quality Control	Adjusted weight
Employment of new marketing and sales staff	0,04	0,04	0,00	0,00	0,00	0,33	0,15	0,52	0,00	0,00	0,00	0,04
Reengineering of marketing division	0,03	0,03	0,00	0,00	0,00	0,52	0,15	0,52	0,00	0,00	0,00	0,03
Determining the requirements about the products	0,03	0,03	0,00	0,00	0,00	0,33	0,52	0,52	0,15	0,15	0,33	0,03
Determining the competitors and performing analysis	0,06	0,06	0,00	0,00	0,15	0,15	0,33	0,33	0,33	0,33	0,15	0,06
Consultancy support about creating a brand	0,02	0,02	0,00	0,00	0,52	0,33	0,00	0,33	0,00	0,00	0,00	0,02
Determining the advantages and disadvantages of the company with respect to the competitors	0,05	0,05	0,00	0,00	0,15	0,15	0,15	0,52	0,33	0,33	0,15	0,05
Attending to the sectoral fairs	0,06	0,06	0,00	0,00	0,00	0,33	0,00	0,33	0,00	0,00	0,00	0,06
Advertisments in the sectoral magazines	0,05	0,05	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,00	0,05
Training support for manufacturing staff	0,06	0,06	0,15	0,33	0,52	0,00	0,33	0,00	0,52	0,00	0,33	0,06
Employment of product engineer	0,07	0,07	0,15	0,33	0,15	0,00	0,33	0,15	0,33	0,15	0,33	0,07
Improvment of machines and equipments	0,06	0,06	0,52	0,52	0,15	0,00	0,33	0,15	0,52	0,33	0,52	0,06
Testing the quality of materials	0,06	0,06	0,52	0,00	0,00	0,00	0,15	0,15	0,52	0,52	0,52	0,06
Efficient calibration	0,07	0,07	0,52	0,15	0,00	0,00	0,00	0,00	0,33	0,33	0,33	0,07
2 shifts a day	0,02	0,02	0,00	0,00	0,00	0,00	0,52	0,52	0,33	0,15	0,33	0,02
Employment of new staff for manufacturing	0,02	0,02	0,00	0,15	0,33	0,00	0,33	0,00	0,33	0,00	0,33	0,02
Research for new technologies	0,07	0,07	0,15	0,15	0,00	0,00	0,15	0,00	0,33	0,15	0,33	0,07
Performing market research	0,06	0,06	0,00	0,00	0,00	0,52	0,00	0,52	0,00	0,33	0,15	0,06
Customer visits	0,07	0,07	0,00	0,00	0,00	0,52	0,00	0,52	0,00	0,00	0,00	0,07
Communication with new customers	0,06	0,06	0,00	0,00	0,00	0,15	0,00	0,52	0,00	0,00	0,15	0,06
Defining the customer needs	0,04	0,04	0,00	0,00	0,00	0,52	0,15	0,52	0,15	0,15	0,33	0,04
Absolute weight			0,01	0,02	0,06	0,08	0,08	0,15	0,07	0,04	0,05	**1,00**
Functional requirement	local		0,02	0,04	0,11	0,15	0,13	0,27	0,13	0,07	0,09	
0,66	global		0,01	0,03	0,07	0,10	0,09	0,18	0,09	0,05	0,06	
Current Status			0,54	0,12	0,34	0,54	0,34	0,12	0,34	0,34	0,34	
Competitor A			0,54	0,34	0,34	0,54	0,34	0,34	0,54	0,34	0,34	
Design Target			0,54	0,34	0,34	0,54	0,54	0,34	0,54	0,34	0,54	
Measurement			linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	
Adjusted weight			0,03	0,05	0,12	0,12	0,13	0,22	0,11	0,09	0,13	**1,00**

Table 6.12 Conversion of processes into functional characteristics

	Priorities	**1,00**																
	Local	Global	Order Receiving	Price proposal	General production planning	Quality plan	Process plan	Control of material amount	Material supply	Material shipment control	Sandblasting	Peeling	Cold drawing	Grinding	Rectification	Prepration of endpoint	Cutting	Polishing
Calibration	0,03	0,03	0,00	0,00	0,15	0,33	0,00	0,00	0,15	0,00	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33
Maintenance	0,05	0,05	0,00	0,00	0,33	0,15	0,15	0,00	0,00	0,00	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33
Training	0,11	0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Customer Relations	0,11	0,11	0,52	0,52	0,15	0,15	0,15	0,00	0,15	0,15	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00
Production	0,14	0,14	0,33	0,33	0,52	0,52	0,52	0,52	0,52	0,15	0,52	0,52	0,52	0,52	0,52	0,52	0,52	0,52
Marketing and order receiving	0,21	0,21	0,52	0,52	0,33	0,33	0,33	0,33	0,33	0,00	0,15	0,33	0,33	0,33	0,33	0,33	0,33	0,33
Warehousing, packaging, shipment	0,11	0,11	0,15	0,00	0,52	0,52	0,52	0,52	0,52	0,52	0,00	0,00	0,00	0,15	0,15	0,15	0,15	0,15
Purchasing	0,09	0,09	0,33	0,15	0,52	0,33	0,33	0,33	0,52	0,52	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15
Quality Control	0,13	0,13	0,15	0,15	0,52	0,15	0,33	0,15	0,33	0,52	0,52	0,52	0,52	0,52	0,52	0,52	0,52	0,52
Absolute weight			0,28	0,25	0,36	0,29	0,30	0,25	0,31	0,21	0,21	0,25	0,25	0,29	0,27	0,27	0,27	0,27
Functional requirement weight	local		0,03	0,03	0,04	0,03	0,03	0,03	0,03	0,02	0,02	0,03	0,03	0,03	0,03	0,03	0,03	0,03
1,00	global		0,03	0,03	0,04	0,03	0,03	0,03	0,03	0,02	0,02	0,03	0,03	0,03	0,03	0,03	0,03	0,03
Current Status			0,12	0,12	0,34	0,34	0,12	0,34	0,12	0,34	0,34	0,34	0,34	0,34	0,34	0,54	0,54	0,34
Competitor A			0,34	0,34	0,34	0,34	0,12	0,34	0,34	0,34	-	-	0,54	-	0,54	-	-	-
Design Target			0,34	0,34	0,54	0,54	0,34	0,54	0,34	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54
Measurement			linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic
Adjusted weight			0,03	0,03	0,04	0,03	0,03	0,03	0,03	0,02	0,02	0,03	0,03	0,03	0,03	0,03	0,03	0,03

Table 6.12 Conversion of processes into functional characteristics (cont.)

																								Planning
	Priorities	**1,00**																						
	Local	Global	Controls in process	Control of finished goods	Determining a nonconforming product	Packaging	Warehousing	Shipment	Evaluation of customer order	Questionnaire analysis	Informing the customer	Diagnosis of the break-down	External maintenance service	Planned maintenance	Supplying the maintenance equipment	Supplier evaluation	Collecting price proposals from the suppliers	Delivering the product/material	Calibration of the measurement equipments	Determining the training needs	Supplying training consultancy	Realization of trainings	Evaluation of trainings	Adjusted weight
Calibration	0,03	0,03	0,00	0,33	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,33	0,33	0,33	0,33	0,00	0,00	0,00	0,52	0,00	0,00	0,00	0,00	0,03
Maintenance	0,05	0,05	0,00	0,15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,52	0,52	0,52	0,15	0,15	0,00	0,33	0,00	0,00	0,00	0,00	0,05
Training	0,11	0,11	0,15	0,15	0,15	0,00	0,00	0,00	0,15	0,15	0,33	0,33	0,33	0,33	0,33	0,15	0,15	0,00	0,00	0,52	0,52	0,52	0,52	0,11
Customer Relations	0,11	0,11	0,00	0,52	0,33	0,33	0,15	0,15	0,52	0,52	0,52	0,15	0,15	0,15	0,15	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,11
Production	0,14	0,14	0,52	0,52	0,33	0,00	0,33	0,33	0,33	0,33	0,15	0,33	0,33	0,33	0,33	0,52	0,52	0,52	0,33	0,33	0,33	0,15	0,15	0,14
Marketing and order receiving	0,21	0,21	0,15	0,33	0,52	0,33	0,15	0,15	0,52	0,52	0,52	0,00	0,00	0,15	0,00	0,15	0,15	0,52	0,00	0,00	0,00	0,00	0,00	0,21
Warehousing, packaging, shipment	0,11	0,11	0,33	0,15	0,33	0,52	0,52	0,52	0,15	0,15	0,33	0,15	0,15	0,33	0,33	0,52	0,15	0,52	0,00	0,00	0,00	0,00	0,00	0,11
Purchasing	0,09	0,09	0,00	0,00	0,15	0,15	0,15	0,15	0,33	0,33	0,15	0,33	0,52	0,33	0,33	0,52	0,52	0,52	0,15	0,33	0,33	0,15	0,00	0,09
Quality Control	0,13	0,13	0,52	0,52	0,52	0,15	0,15	0,15	0,33	0,33	0,15	0,33	0,33	0,15	0,15	0,52	0,15	0,33	0,33	0,33	0,33	0,15	0,52	0,13
Absolute weight			0,23	0,32	0,36	0,20	0,19	0,19	0,32	0,32	0,30	0,23	0,25	0,26	0,22	0,30	0,21	0,35	0,14	0,18	0,18	0,11	0,15	
Functional requirement weight	local		0,02	0,03	0,04	0,02	0,02	0,02	0,03	0,03	0,03	0,02	0,03	0,03	0,02	0,03	0,02	0,04	0,01	0,02	0,02	0,01	0,02	
1,00	global		0,02	0,03	0,04	0,02	0,02	0,02	0,03	0,03	0,03	0,02	0,03	0,03	0,02	0,03	0,02	0,04	0,01	0,02	0,02	0,01	0,02	
Current Status			0,54	0,54	0,34	0,34	0,34	0,34	0,34	0,34	0,54	0,34	0,12	0,34	0,34	0,34	0,34	0,54	0,54	0,34	0,54	0,34	0,34	
Competitor A			0,54	0,54	0,34	0,34	0,54	0,54	0,34	0,34	0,34	0,34	0,12	0,34	0,34	0,34	0,34	0,54	0,34	0,34	0,34	0,34	0,34	
Design Target			0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,34	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	
Measurement			linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	linguistic	
Adjusted weight			0,02	0,03	0,04	0,02	0,02	0,02	0,03	0,03	0,03	0,02	0,03	0,03	0,02	0,03	0,02	0,04	0,01	0,02	0,02	0,01	0,02	

Table 6.13 Conversion of functional characteristics into informational characteristics

Priorities 1,00	Local	Global	Maintenance followup card	Equipment list for calibration	Calibration followup card	Trainning plan	Training attendance form	Staff followup card	Training evaluation form	Corrective and preventive activities procedure	Customer satisfaction questionnaire	Customer complaints followup and evaluation form	Nonconforming product evaluation fo	Product refusal form	Nonconforming product evaluation procedure
Order Receiving	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15	0,15	0,00	0,00	0,00
Price proposal	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
General production planning	0,04	0,04	0,33	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,33
Quality plan	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,33	0,33	0,33
Process plan	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Control of material amount	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Material supply	0,03	0,03	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Material shipment control	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15	0,15	0,33
Sandblasting	0,02	0,02	0,33	0,15	0,15	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00
Peeling	0,03	0,03	0,33	0,15	0,15	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00
Cold drawing	0,03	0,03	0,33	0,15	0,15	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00
Grinding	0,03	0,03	0,33	0,15	0,15	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00
Rectification	0,03	0,03	0,33	0,15	0,15	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00
Prepration of endpoint	0,03	0,03	0,33	0,15	0,15	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00
Cutting	0,03	0,03	0,33	0,15	0,15	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00
Polisaj	0,03	0,03	0,33	0,15	0,15	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00
Controls in process	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,15	0,15	0,15
Control of finished goods	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,52	0,15	0,52
Determining a nonconforming product	0,04	0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,33	0,00	0,00	0,52	0,15	0,52
Packaging	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,33	0,00	0,00	0,52	0,00	0,33
Warehousing	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,33	0,00	0,00	0,52	0,00	0,33
Shipment	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,33	0,00	0,00	0,52	0,00	0,33
Evaluation of customer order	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15	0,33	0,33	0,00	0,33	0,00
Questionnaire analysis	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,52	0,00	0,15	0,00
Informing the customer	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,33	0,52	0,00	0,33	0,00
Diagnosis of the break-down	0,02	0,02	0,52	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
External maintenance service	0,03	0,03	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Planned maintenance	0,03	0,03	0,52	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Supplying the maintenance equipmentq	0,02	0,02	0,52	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Supplier evaluation	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Collecting price proposals from the suppliers	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Delivering the product/material	0,04	0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Calibration of the measurement equipments	0,01	0,01	0,00	0,52	0,52	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Determining the training needs	0,02	0,02	0,00	0,00	0,00	0,52	0,52	0,52	0,52	0,00	0,00	0,00	0,00	0,00	0,00
Supplying training consultancy	0,02	0,02	0,00	0,00	0,00	0,52	0,00	0,00	0,33	0,00	0,00	0,00	0,00	0,00	0,00
Realization of trainings	0,01	0,01	0,00	0,00	0,00	0,00	0,52	0,52	0,33	0,00	0,00	0,00	0,00	0,00	0,00
Evaluation of trainings	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,00	0,00	0,00	0,00
Absolute weight			0,15	0,04	0,05	0,02	0,02	0,02	0,03	0,10	0,04	0,05	0,11	0,06	0,09
Functional requirement weight	local		0,05	0,01	0,02	0,01	0,01	0,01	0,01	0,03	0,01	0,02	0,04	0,02	0,03
1,00	global		0,05	0,01	0,02	0,01	0,01	0,01	0,01	0,03	0,01	0,02	0,04	0,02	0,03

Table 6.13 Conversion of functional characteristics into informational characteristics (cont.)

	Local	Global	Production followup and control card	Final quality control report	Nonconforming product evaluation report	Sales proposal and order form	Quality plan for production	Inventory list	Quality control report for material	Material entrance receipt	Instruction for material acceptance	Handling,warehousing and packaging instructions	Supplier evaluation form	Purchasing proposal form	Supplier list
Priorities		1,00													
Order Receiving	0,03	0,03	0,00	0,00	0,00	0,52	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Price proposal	0,03	0,03	0,00	0,00	0,00	0,52	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
General production planning	0,04	0,04	0,52	0,52	0,52	0,52	0,33	0,52	0,15	0,15	0,00	0,33	0,00	0,15	0,15
Quality plan	0,03	0,03	0,33	0,33	0,33	0,00	0,52	0,00	0,33	0,15	0,15	0,00	0,00	0,00	0,00
Process plan	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Control of material amount	0,03	0,03	0,15	0,00	0,00	0,00	0,00	0,33	0,52	0,52	0,33	0,00	0,00	0,33	0,00
Material supply	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,52	0,33	0,33	0,52	0,52	0,52	0,52	0,52
Material shipment control	0,02	0,02	0,00	0,33	0,33	0,00	0,33	0,00	0,52	0,52	0,52	0,33	0,00	0,00	0,00
Sandblasting	0,02	0,02	0,52	0,15	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Peeling	0,03	0,03	0,52	0,15	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Cold drawing	0,03	0,03	0,52	0,15	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Grinding	0,03	0,03	0,52	0,15	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Rectification	0,03	0,03	0,52	0,15	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Prepration of endpoint	0,03	0,03	0,52	0,15	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Cutting	0,03	0,03	0,52	0,15	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Polisaj	0,03	0,03	0,52	0,15	0,00	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Controls in process	0,02	0,02	0,33	0,52	0,33	0,00	0,52	0,00	0,33	0,00	0,00	0,00	0,00	0,00	0,00
Control of finished goods	0,03	0,03	0,33	0,52	0,52	0,00	0,52	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00
Determining a nonconforming product	0,04	0,04	0,15	0,33	0,52	0,00	0,33	0,00	0,33	0,00	0,00	0,00	0,00	0,00	0,00
Packaging	0,02	0,02	0,00	0,33	0,52	0,00	0,00	0,15	0,52	0,52	0,52	0,52	0,00	0,00	0,00
Warehousing	0,02	0,02	0,00	0,33	0,52	0,00	0,00	0,15	0,52	0,52	0,52	0,52	0,00	0,00	0,00
Shipment	0,02	0,02	0,00	0,33	0,52	0,00	0,00	0,15	0,52	0,52	0,52	0,52	0,00	0,00	0,00
Evaluation of customer order	0,03	0,03	0,00	0,00	0,00	0,52	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Questionnaire analysis	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Informing the customer	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Diagnosis of the break-down	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
External maintenance service	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,33	0,33
Planned maintenance	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Supplying the maintenance equipmentq	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Supplier evaluation	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,33	0,52
Collecting price proposals from the suppliers	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,52
Delivering the product/material	0,04	0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,52	0,52	0,52	0,33	0,33	0,33
Calibration of the measurement equipments	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Determining the training needs	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Supplying training consultancy	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,33	0,00
Realization of trainings	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Evaluation of trainings	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Absolute weight			0,18	0,13	0,12	0,07	0,11	0,07	0,13	0,10	0,09	0,09	0,05	0,08	0,07
Functional requirement weight	local		0,06	0,04	0,04	0,02	0,04	0,02	0,04	0,03	0,03	0,03	0,02	0,03	0,02
1,00	global		0,06	0,04	0,04	0,02	0,04	0,02	0,04	0,03	0,03	0,03	0,02	0,03	0,02

Table 6.13 Conversion of functional characteristics into informational characteristics (cont.)

	Local	Global	Internal communication form	Internal order form	Control plan form for acceptance of materials	Nonconforming product customer repo	Nonconforming product followup form	List for materials tobe analyzed	Monthly report for materials enterance to warehouse	Acceptance report for outsourced materials/products	Outsourced production followup card	Manufacturing process followup card	Chrome plating (outsource) followup form	Final quality control plan for products	Adjusted weight
Priorities		**1,00**													
Order Receiving	0,03	0,03	0,15	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Price proposal	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
General production planning	0,04	0,04	0,15	0,33	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,52	0,33	0,04
Quality plan	0,03	0,03	0,00	0,00	0,33	0,00	0,00	0,15	0,15	0,15	0,15	0,15	0,15	0,52	0,03
Process plan	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Control of material amount	0,03	0,03	0,33	0,33	0,33	0,00	0,00	0,33	0,52	0,52	0,52	0,15	0,15	0,33	0,03
Material supply	0,03	0,03	0,33	0,33	0,33	0,00	0,00	0,33	0,33	0,33	0,00	0,00	0,00	0,15	0,03
Material shipment control	0,02	0,02	0,00	0,00	0,52	0,15	0,15	0,15	0,52	0,52	0,33	0,00	0,00	0,52	0,02
Sandblasting	0,02	0,02	0,15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,02
Peeling	0,03	0,03	0,15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,03
Cold drawing	0,03	0,03	0,15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,03
Grinding	0,03	0,03	0,15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,03
Rectification	0,03	0,03	0,15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,03
Prepration of endpoint	0,03	0,03	0,15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,03
Cutting	0,03	0,03	0,15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,03
Polisaj	0,03	0,03	0,15	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,00	0,03
Controls in process	0,02	0,02	0,00	0,00	0,00	0,15	0,15	0,00	0,00	0,00	0,52	0,52	0,15	0,52	0,02
Control of finished goods	0,03	0,03	0,00	0,00	0,00	0,33	0,33	0,00	0,00	0,00	0,52	0,52	0,15	0,52	0,03
Determining a nonconforming product	0,04	0,04	0,00	0,00	0,00	0,33	0,33	0,00	0,00	0,00	0,33	0,33	0,33	0,33	0,04
Packaging	0,02	0,02	0,33	0,33	0,33	0,33	0,15	0,00	0,15	0,15	0,15	0,00	0,00	0,33	0,02
Warehousing	0,02	0,02	0,33	0,33	0,33	0,33	0,15	0,00	0,15	0,15	0,15	0,00	0,00	0,33	0,02
Shipment	0,02	0,02	0,33	0,33	0,33	0,33	0,15	0,00	0,15	0,15	0,15	0,00	0,00	0,33	0,02
Evaluation of customer order	0,03	0,03	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Questionnaire analysis	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Informing the customer	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Diagnosis of the break-down	0,02	0,02	0,33	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02
External maintenance service	0,03	0,03	0,33	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Planned maintenance	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Supplying the maintenance equipmentq	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02
Supplier evaluation	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Collecting price proposals from the suppliers	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02
Delivering the product/material	0,04	0,04	0,15	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,04
Calibration of the measurement equipments	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01
Determining the training needs	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02
Supplying training consultancy	0,02	0,02	0,15	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02
Realization of trainings	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01
Evaluation of trainings	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02
Absolute weight			0,11	0,16	0,07	0,05	0,04	0,03	0,05	0,05	0,08	0,19	0,05	0,12	**3,04**
Functional requirement weight	local		0,04	0,05	0,02	0,02	0,01	0,01	0,02	0,02	0,03	0,06	0,02	0,04	**1,00**
1,00	global		0,04	0,05	0,02	0,02	0,01	0,01	0,02	0,02	0,03	0,06	0,02	0,04	**1,00**

Table 6.14 Conversion of functional characteristics into organizational characteristics

	Prioritie	**1,00**										
	Local	Global	Manufacturing staff	Production supervisor	Productionand Sales manager	Quality System and Quality Control Manager	Quality Control Staff	Foreman	Material acceptance and storage supervisor	Raw material purchasing supervisor	Consumption materials purchasing supervisor	Adjusted weight
Order ceceiving	0,03	0,03	0,00	0,00	0,52	0,15	0,00	0,00	0,15	0,52	0,15	0,03
Price proposal	0,03	0,03	0,00	0,00	0,52	0,15	0,00	0,00	0,00	0,52	0,15	0,03
General production planning	0,04	0,04	0,15	0,52	0,33	0,15	0,00	0,00	0,15	0,33	0,52	0,04
Quality plan	0,03	0,03	0,15	0,52	0,33	0,52	0,00	0,00	0,15	0,33	0,00	0,03
Process plan	0,03	0,03	0,15	0,52	0,33	0,15	0,00	0,00	0,15	0,33	0,52	0,03
Control of material amount	0,03	0,03	0,33	0,33	0,00	0,15	0,33	0,00	0,52	0,00	0,15	0,03
Material supply	0,03	0,03	0,15	0,00	0,52	0,15	0,00	0,00	0,00	0,52	0,00	0,03
Peeling	0,03	0,03	0,52	0,15	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,03
Cold drawing	0,03	0,03	0,52	0,15	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,03
Grinding	0,03	0,03	0,52	0,15	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,03
Rectification	0,03	0,03	0,52	0,15	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,03
Prepration of endpoint	0,03	0,03	0,52	0,15	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,03
Cutting	0,03	0,03	0,52	0,15	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,03
Polisaj	0,03	0,03	0,52	0,15	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,03
Controls in process	0,02	0,02	0,52	0,15	0,00	0,52	0,00	0,00	0,00	0,00	0,00	0,02
Control of finished goods	0,03	0,03	0,52	0,15	0,00	0,52	0,52	0,00	0,15	0,00	0,00	0,03
Determining a nonconforming product	0,04	0,04	0,52	0,15	0,00	0,52	0,52	0,00	0,15	0,00	0,00	0,04
Packaging	0,02	0,02	0,15	0,15	0,15	0,33	0,52	0,00	0,52	0,15	0,33	0,02
Warehousing	0,02	0,02	0,00	0,15	0,15	0,33	0,33	0,00	0,52	0,15	0,33	0,02
Shipment	0,02	0,02	0,00	0,15	0,15	0,33	0,33	0,00	0,52	0,15	0,00	0,02
Evaluation of customer order	0,03	0,03	0,00	0,00	0,52	0,15	0,33	0,00	0,52	0,52	0,00	0,03
Questionnaire analysis	0,03	0,03	0,00	0,00	0,33	0,52	0,00	0,00	0,00	0,33	0,00	0,03
Informing the customer	0,03	0,03	0,00	0,33	0,52	0,33	0,00	0,00	0,33	0,52	0,33	0,03
Diagnosis of the break-down	0,02	0,02	0,52	0,15	0,15	0,15	0,00	0,52	0,00	0,15	0,15	0,02
External maintenance service	0,03	0,03	0,00	0,52	0,33	0,15	0,00	0,52	0,00	0,33	0,52	0,03
Planned maintenance	0,03	0,03	0,52	0,33	0,00	0,33	0,00	0,52	0,00	0,00	0,33	0,03
Supplying the maintenance equipment	0,02	0,02	0,00	0,52	0,33	0,15	0,00	0,52	0,00	0,33	0,52	0,02
Supplier evaluation	0,03	0,03	0,00	0,33	0,33	0,52	0,00	0,00	0,00	0,33	0,33	0,03
Collecting price proposals from the suppliers	0,02	0,02	0,00	0,33	0,33	0,33	0,00	0,00	0,00	0,33	0,33	0,02
Delivering the product/material	0,04	0,04	0,00	0,00	0,33	0,15	0,00	0,00	0,52	0,33	0,15	0,04
Calibration of the measurement equipments	0,01	0,01	0,52	0,33	0,00	0,52	0,52	0,52	0,00	0,00	0,15	0,01
Determining the training needs	0,02	0,02	0,00	0,15	0,15	0,52	0,00	0,00	0,00	0,15	0,15	0,02
Supplying training consultancy	0,02	0,02	0,00	0,00	0,15	0,52	0,00	0,00	0,00	0,15	0,15	0,02
Realization of trainings	0,01	0,01	0,00	0,15	0,15	0,52	0,00	0,00	0,00	0,15	0,15	0,01
Evaluation of trainings	0,02	0,02	0,00	0,33	0,15	0,52	0,00	0,00	0,00	0,15	0,15	0,02
Absolute weight			0,24	0,20	0,20	0,29	0,10	0,06	0,14	0,20	0,15	1,57
Functional requirement weight	local		0,15	0,13	0,13	0,18	0,07	0,04	0,09	0,13	0,09	**1,00**
1,00	global		0,15	0,13	0,13	0,18	0,07	0,04	0,09	0,13	0,09	**1,00**

Table 6.15 Conversion of functional characteristics into resource characteristics

	Priorities	**1,00**								
	Local	Global	Sandblasting machine	Grinding1	Grinding2	Grinding3	Rectification1	Rectification2	Rectification3	Rectification4
Order Receiving	0,03	0,03	0,15	0,33	0,33	0,33	0,15	0,15	0,15	0,15
Price proposal	0,03	0,03	0,15	0,33	0,33	0,33	0,15	0,15	0,15	0,15
General production planning	0,04	0,04	0,33	0,52	0,52	0,52	0,52	0,52	0,52	0,52
Quality plan	0,03	0,03	0,52	0,52	0,52	0,52	0,52	0,52	0,52	0,52
Process plan	0,03	0,03	0,52	0,52	0,52	0,52	0,52	0,52	0,52	0,52
Control of material amount	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Material supply	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Material shipment control	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Sandblasting	0,02	0,02	0,52	0,33	0,33	0,33	0,00	0,00	0,00	0,00
Peeling	0,03	0,03	0,00	0,33	0,33	0,33	0,00	0,00	0,00	0,00
Cold drawing	0,03	0,03	0,52	0,33	0,33	0,33	0,00	0,00	0,00	0,00
Grinding	0,03	0,03	0,00	0,52	0,52	0,52	0,00	0,00	0,00	0,00
Rectification	0,03	0,03	0,52	0,33	0,33	0,33	0,00	0,00	0,00	0,00
Prepration of endpoint	0,03	0,03	0,00	0,33	0,33	0,33	0,00	0,00	0,00	0,00
Cutting	0,03	0,03	0,33	0,33	0,33	0,33	0,00	0,00	0,00	0,00
Polisaj	0,03	0,03	0,33	0,33	0,33	0,33	0,00	0,00	0,00	0,00
Controls in process	0,02	0,02	0,52	0,52	0,52	0,52	0,52	0,52	0,52	0,52
Control of finished goods	0,03	0,03	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15
Determining a nonconforming product	0,04	0,04	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15
Packaging	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Warehousing	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Shipment	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Evaluation of customer order	0,03	0,03	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15
Questionnaire analysis	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Informing the customer	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Diagnosis of the break-down	0,02	0,02	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33
External maintenance service	0,03	0,03	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15
Planned maintenance	0,03	0,03	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33
Supplying the maintenance equipment	0,02	0,02	0,33	0,33	0,33	0,33	0,33	0,33	0,33	0,33
Supplier evaluation	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Determining the training needs	0,02	0,02	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15
Supplying training consultancy	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Realization of trainings	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Evaluation of trainings	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Absolute weight			0,17	0,21	0,21	0,21	0,12	0,12	0,12	0,12
Functional requirement weight	local		0,07	0,08	0,08	0,08	0,05	0,05	0,05	0,05
1,00	global		0,07	0,08	0,08	0,08	0,05	0,05	0,05	0,05
min			44	43	43	43	38	202	104	104
max			6088	2138	2138	2138	3849	3267	9812	9812
Measurement			kg/hr	kg/hr	kg/hr	kg/hr	kg/hr	kg/hr	kg/hr	kg/hr

Table 6.15 Conversion of functional characteristics into resource characteristics cont.)

	Priorities	1,00									
	Local	Global	Rectification5	Cold Drawing1	Cold Drawing2	Peeling	Polishing	Cutting	Warehouse	Measurement equipments	**Adjusted weight**
Order Receiving	0,03	0,03	0,15	0,33	0,33	0,33	0,33	0,33	0,52	0,00	0,03
Price proposal	0,03	0,03	0,15	0,33	0,33	0,33	0,33	0,33	0,52	0,00	0,03
General production planning	0,04	0,04	0,52	0,52	0,52	0,52	0,52	0,52	0,15	0,00	0,04
Quality plan	0,03	0,03	0,52	0,52	0,52	0,52	0,52	0,52	0,15	0,52	0,03
Process plan	0,03	0,03	0,52	0,52	0,52	0,52	0,52	0,52	0,15	0,52	0,03
Control of material amount	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,03
Material supply	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,00	0,03
Material shipment control	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,33	0,02
Sandblasting	0,02	0,02	0,00	0,52	0,52	0,00	0,00	0,00	0,00	0,00	0,02
Peeling	0,03	0,03	0,00	0,00	0,00	0,52	0,00	0,00	0,00	0,00	0,03
Cold drawing	0,03	0,03	0,00	0,52	0,52	0,00	0,00	0,00	0,00	0,00	0,03
Grinding	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Rectification	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Prepration of endpoint	0,03	0,03	0,00	0,52	0,52	0,00	0,00	0,00	0,00	0,00	0,03
Cutting	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Polisaj	0,03	0,03	0,00	0,00	0,00	0,00	0,52	0,15	0,00	0,00	0,03
Controls in process	0,02	0,02	0,52	0,52	0,52	0,52	0,52	0,52	0,00	0,52	0,02
Control of finished goods	0,03	0,03	0,15	0,15	0,15	0,33	0,33	0,52	0,00	0,52	0,03
Determining a nonconforming product	0,04	0,04	0,15	0,15	0,15	0,15	0,15	0,15	0,00	0,52	0,04
Packaging	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,15	0,02
Warehousing	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,15	0,02
Shipment	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,52	0,15	0,02
Evaluation of customer order	0,03	0,03	0,15	0,33	0,33	0,33	0,33	0,33	0,15	0,00	0,03
Questionnaire analysis	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Informing the customer	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Diagnosis of the break-down	0,02	0,02	0,33	0,33	0,33	0,33	0,33	0,33	0,00	0,15	0,02
External maintenance service	0,03	0,03	0,15	0,15	0,15	0,15	0,15	0,15	0,00	0,00	0,03
Planned maintenance	0,03	0,03	0,33	0,33	0,33	0,33	0,33	0,33	0,00	0,00	0,03
Supplying the maintenance equipment	0,02	0,02	0,33	0,33	0,33	0,33	0,33	0,33	0,15	0,00	0,02
Supplier evaluation	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,03
Determining the training needs	0,02	0,02	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,02
Supplying training consultancy	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02
Realization of trainings	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01
Evaluation of trainings	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02
Absolute weight			0,12	0,18	0,18	0,16	0,16	0,16	0,14	0,13	2,52
Functional requirement weight	local		0,05	0,07	0,07	0,06	0,06	0,06	0,06	0,05	**1,00**
1,00	global		0,05	0,07	0,07	0,06	0,06	0,06	0,06	0,05	**1,00**
min			104	94	46	44	*	*	300	*	
max			9812	7966	3938	1739	*	*	500	*	
Measurement			kg/hr	kg/hr	kg/hr	kg/hr	kg/hr	kg/hr	tone	*	

The second matrix of Enterprise-QFD, Table 6.11, indicates the relationships between process goals and business processes. One goal can cover more than one business process or any of the processes can serve for more than one process goals. Table 6.11 helps the enterprise engineer to see these relations.

In the third matrix of Enterprise-QFD, Table 6.12, business processes are broken down into functional requirements within the company. The relationships are then shown between the business processes and the functional requirements. This phase is another milestone where the interactions between the processes and functional characteristics, and conflicts (if exist) between each couple of functional characteristics are observed. The remaining matrices, from Table 6.13 to Table 6.15, convert functional requirements into informational requirements, resource requirements, and organizational requirements, respectively. The relationship matrix for informational requirements in Table 6.13 can be used to design databases through relationships between the objects and functional requirements.

The fourth matrix of Enterprise-QFD can be developed to convert the modelling constructs if any reference architecture is selected for the implementation. The evaluations should also be cross-checked with different staff for validation. The adjusted importance weights of the design part show the design characteristics with their priorities among all design items. They also support the enterprise engineer in modelling the requirements.

The matrix explanations mention about a conversion. The term conversion means that the input weights are taken from the previous matrix in the sequence, and evaluated according to the relationship values between the input characteristics by using the scale developed in Table 6.6 through 6.9. Finally, the normalized weights of the characteristics that are given in the column of the matrix are obtained by applying the equations from 5.1 to 5.4. The columns are defined by the enterprise engineer and other staff in modelling team and weights are calculated after all evaluations are obtained. Each matrix presents the level of relationships between columns and rows, thus, when a change is needed in the enterprise, all matrices can be used to see how this change will affect the processes, functional, informational, resource, and organizational characteristics where the weights indicate the level of the relationships. As a result, Enterprise-QFD yields an infrastructure for sustainability of the enterprise model by providing a detailed database with importance values and relationships for all modelling components concerning the enterprise goals.

Some other parts can be added to the matrices for further evaluations about company or competitors as long as the sufficient data exist. Besides, an optimization method can be applied to find the relationships or design targets. For example, resource optimization can be applied to the matrix in Table 6.15 where the functional requirements are converted into resource requirements.

The evaluations and relationships defined in the matrix might indicate that any process, goal, functional characteristic, or information object is related to the others more significantly. Therefore, any minor change in a design feature might affect some features more than the others. Enterprise-QFD supports enterprise engineer to visualize relationships to see the interactions and the affects of changes on other processes. The conflicts or correlations among characteristics or goals can also be visualized through the roof of the house.

Enterprise-QFD has been implemented in the SME mentioned in section 6.3.1and successful findings have been observed. For example, during the preparation phase voice of the users are analyzed and major concepts are retrieved from their declarations. These declarations are compared with the predetermined long term enterprise goals to avoid from the goals not supporting requirements or vice versa. Second advantage arises during the evaluations through the integrated matrix given in Figure 5.2. At the end of each evaluation, importance or weights of the characteristics (goals, processes, modelling characteristics) are obtained. Therefore, outputs can show which characteristic is critical, or completely not important by the point of view of the users. Enterprise goals or process goals may have very low value. Enterprise engineer has a chance to see this reality and to re-evaluate them. In addition to those advantages, because each output of a matrix is the input of the following one, the evaluations carry the relationships from the goals to the elementary enterprise modelling objects.

The requirement analysis model developed for discovering the enterprise requirements extends the Modern QFD. Each preparation phase of Modern QFD is transformed according to the enterprise modelling concepts. The tables used for gathering the user's requirements and problems are modified cancelling the fields which are not related with enterprise modelling, and some fields are added required for enterprise issues. Measurements and priorities are obtained based on Fuzzy-AHP instead of classical AHP concerning the uncertainties within and around enterprises.

Modern QFD concerns the customer requirements to be transformed into product/service characteristics. For this purpose, some evaluations are performed through matrices with mathematical evaluations after the detailed preparation phase. Enterprise modelling issues are different from the design of product / service, thus the content and the sequential process of evaluation matrices are developed from scratch for enterprise modelling, and finally the original matrix group and analysis model obtained, namely, Enterprise-QFD. The matrices are represented with the required sequence in Figure 5.2.

Enterprise-QFD has many advantages for an enterprise engineer to manage the enterprise modelling process if all integrated tools can be applied properly and in compatible with the related systems. Enterprise modelling and Enterprise-QFD should be in parallel with the strategic decision making level. Strategic plans include detailed analysis about enterprise goals, strategies, customers and stakeholders, and competitive status of the company. These predefined statements can be inputs to the requirements analysis. Processes may also be predefined in different systems, e.g. in ISO 9000 documents. Process definition is not easy to be completed properly in a short time and may be taken as a particular part of enterprise modelling. The process definitions can directly contribute the Enterprise-QFD process, if they have been completed before the enterprise modelling as long as they are reconsidered during the enterprise modelling activities. The common definitions in strategic planning, enterprise modelling, and other implementations in the company should be guaranteed that they are compatible and not in contradiction. Otherwise, Enterprise-QFD cannot be beneficiary in those potential contradictions and unnecessary repetitions may result in a disadvantage for the enterprise modelling process.

Each step in Enterprise-QFD requires group decision making activities and qualitative evaluations, thus Enterprise-QFD can easily be affected by group dynamics. Dominant people may insist on their own preferences and they may keep on persuading the other during the evaluations. Top management should provide such a group platform to manage the evaluation meetings where each group member can propose his thoughts without being under pressure. Another disadvantage may arise during the qualitative evaluations throughout the linguistic variables. Customer Voice Table and Maximum Value Table statements may require general QFD training before using them because of the difficulty in retrieving the clarified items from the needs. Besides, the scale of measurement which is developed for the evaluations should be calculated with respect to the perceptions of people in the enterprise. The quantitative measures of linguistic variables should be shared with the people in the decision group and should be verified that the scale values represent their perceptions. Management and the other staff in the decision group within an enterprise should consider these conditions and potential disadvantages while implementing Enterprise-QFD.

The standard formulation is presented and used for the calculations in the QFD matrices. All the scales of measurement used for evaluations are in the normalized form, thus the evaluation values are defined by ratio scale numbers between 0 and 1. However, “competitive improvement ratio” that is defined by equation 5.2 is the division of “position plan for improvement” by “current status” of a particular statement in a matrix and this ratio is aggregated to the calculations without being normalized. It would be more consistent if the normalized form of the competitive improvement ratio is employed during the calculations in each matrix so that each scale of measurement is defined as a ratio between 0 and 1.

Enterprise-QFD is a detailed method including many phases and different techniques during matrix calculations. Enterprise-QFD may be a time consuming process, if the process is handled by an employee manually who does not have deep knowledge about QFD. The author has overcome this disadvantage by developing a toolbox application working on MS-Excel, which is introduced in Section 6.5. The phases of Enterprise-QFD requiring evaluations and calculations have been transformed into tables in MS-Excel, and user forms have been designed for sequential evaluations where the results are calculated automatically and recorded as structured reports in different sheets. One of the most complex and difficult type of enterprise modelling is to model an enterprise in which computer integrated manufacturing components are implemented. The complexity is due to the need for a complete integration. All relationships between process goals and processes, functional characteristics and processes, functional characteristics and the other modelling constructs, i.e., resource, information, and organization should be analyzed and clearly defined. The most common reference model is CIMOSA, which is used in enterprises managing computer integrated manufacturing. Enterprise-QFD, as a requirement analysis approach which is proposed especially for the enterprise engineers, can manage all requirement analysis and definition phase of enterprise modelling. Enterprise-QFD supports all modelling components of CIMOSA within the series of detailed matrices. The matrix in Table 6.12 defines relationships and importance levels of processes and functional characteristics of the enterprise model which corresponds to the functional view of CIMOSA. Then, these functional characteristics are evaluated with respect to the informational, resource, and organizational characteristics, respectively given in Table 6.13 to 6.15. These characteristics also correspond to the related views of CIMOSA which is one of the most complex reference architectures among the other architectures. The ease of use of the proposed approach is approved during the transformation process from requirements analysis results into CIMOSA constructs (see section 6.6). The findings in last four matrices are clarified and ready to be used in a particular enterprise reference architecture. This transformation process is faster than the expected time period and the items in the matrices are well matched with the reference components.

Thus, the outputs of the requirement analysis are directly the inputs of the enterprise reference architectures.

6.4 A Simple Software Toolbox for Enterprise-QFD

Some formulations and calculations are needed during the construction and fulfilment of the matrices such as priority calculations using Fuzzy-AHP and weighted sums of each part in matrices to complete the phase through the formulations that are given in chapter 5. Therefore, enterprise engineers managing the modelling process should completely learn how to calculate each sub model within the matrices, and also have to perform all operations repeatedly for any minor change.

A toolbox has been developed to address the abovementioned issues. The algorithms are created to run the Fuzzy-AHP, and tables are constructed including user forms for evaluations. Prioritization of enterprise goals and matrix evaluations of Enterprise-QFD are coded in VBA in MS-EXCEL. First of all, a menu is added on the menu bar including the calculations in Enterprise-QFD step by step. Then, a new worksheet is designed for each phase of implementation. The first screen is developed to introduce the Enterprise-QFD, the phases which should be completed before using the toolbox, and related explanations for evaluations. Figure 6.5 represents the first screen and the content of the Enterprise-QFD with the sequence of the steps to be performed. This toolbox can be used in two ways; users can prefer continuing the menu on the top of the MS-Excel or use “next” buttons on the bottom of each screen for the further phase.

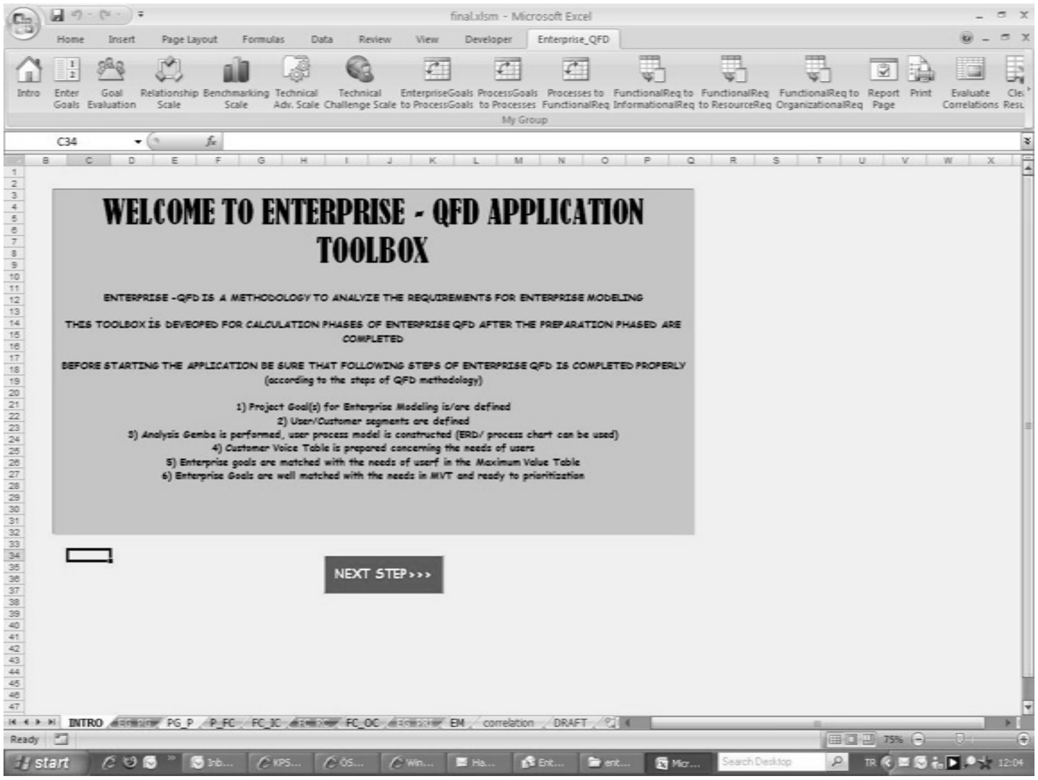


Figure 6.5 Introduction page and menu of enterprise-QFD.

As shown in Figure 6.5, the MVT determines the enterprise goals matching them with the needs just before the calculations are started. After all enterprise goals are verified by the user needs, these goals are prioritized by using Fuzzy-AHP in Enterprise-QFD. This is the starting point of the toolbox and provides the first advantage because of the complexity in Fuzzy-AHP calculations. Fuzzy-AHP consists of special algorithms with triangular fuzzy numbers; thus this phase would need an expert user, unless this simple toolbox exists. Figure 6.6 and 6.7 are the user forms which are generated to access the enterprise goals and evaluate them by pairwise comparisons using the scale given in Figure 6.7.

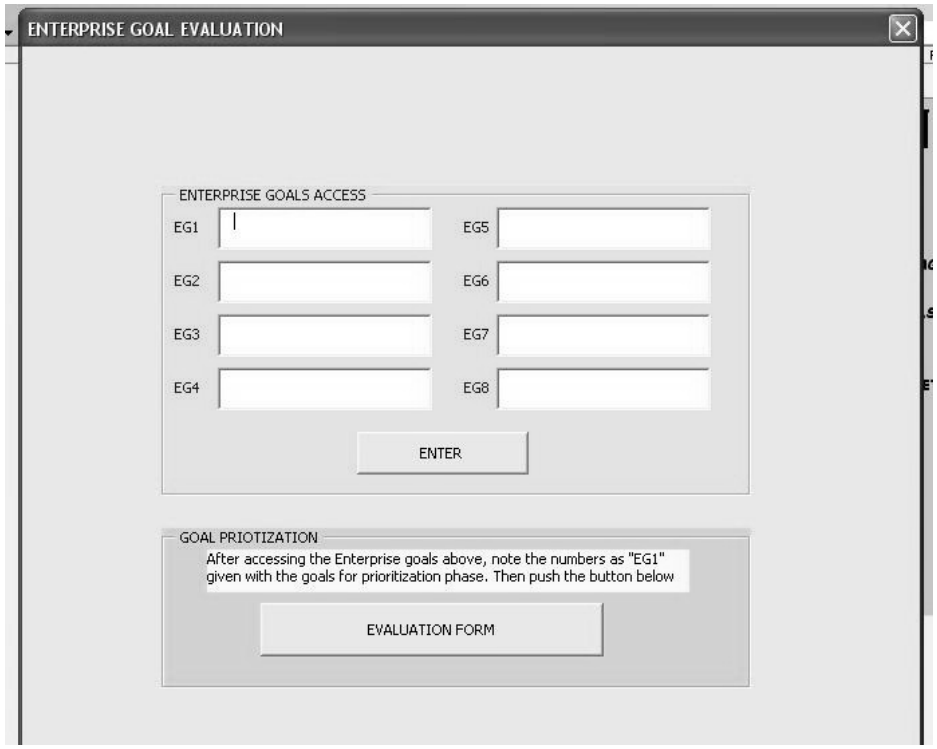


Figure 6.6 User form for enterprise goal access.

Fuzzy-AHP is used for two different purposes in Enterprise-QFD. The first one is the prioritization of the enterprise goals and the second one is to obtain a ratio scale for evaluations in the matrices. Therefore, the content of Figure 6.7 is also used to obtain measures which are given in Figure 6.8. The results are recorded in a sheet to be used in further calculations and reports.

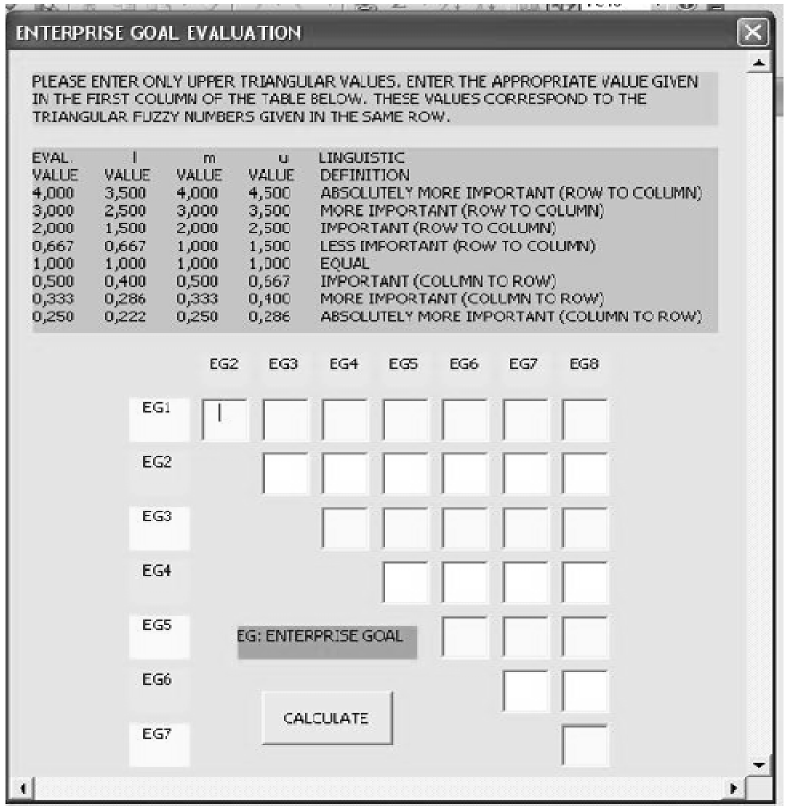


Figure 6.7 Evaluation of enterprise goals.

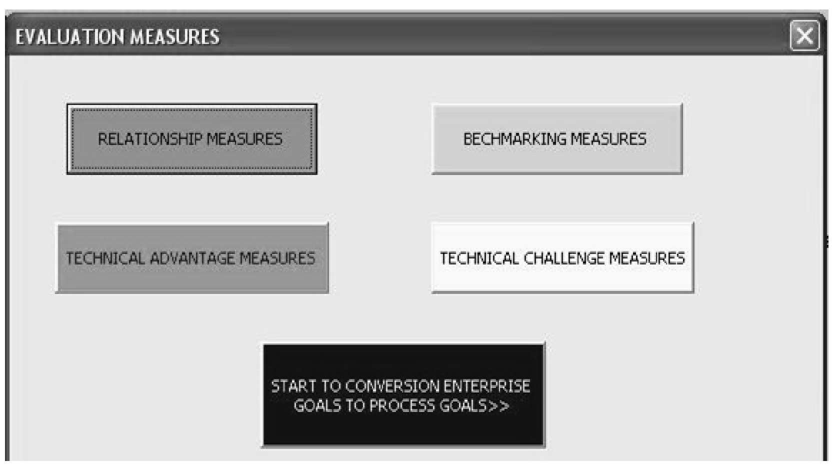


Figure 6.8 Access to evaluation measures.

So far, enterprise goals are determined and accessed to the toolbox through the user forms, and importance values of enterprise goals and evaluation measures are calculated. Input values are completed for the evaluation and transformation processes which will be performed within the matrices. For this purpose, a dynamic worksheet is designed utilizing the excel worksheet advantages, by adding command buttons and user forms to select the evaluation measures so that the real scale values are loaded on the related fields of the worksheet. In each worksheet some buttons are added to load the input values, transform the outputs from the current matrix into the next matrix as the input values, and transfer the current results to the result page.

The worksheets are designed with respect to the colours of the regions where each region corresponds to the different part of the QFD matrix. For example, light blue region in Figure 6.9 represents the relationship matrix part. Whenever one of the cells in the light blue region is selected, a user form appears including relationship measures. If pink region is selected, benchmarking measures appear to be used as a specifically defined user form. One of the matrices transforming enterprise goals into process goals is given in Figure 6.9. The user forms generated for evaluation of each part of the matrix have different evaluation expressions. Whenever an expression is selected and approved by pushing the “ok” button, the related evaluation value is automatically placed in the selected cell.

Enterprise-QFD performs the transformation process via series of matrices as explained in section 6.3. Different transformations are handled in each of the matrix using the same structure in Figure 6.9. In other words, the structures of the matrices are the same as the others accomplishing different tasks.

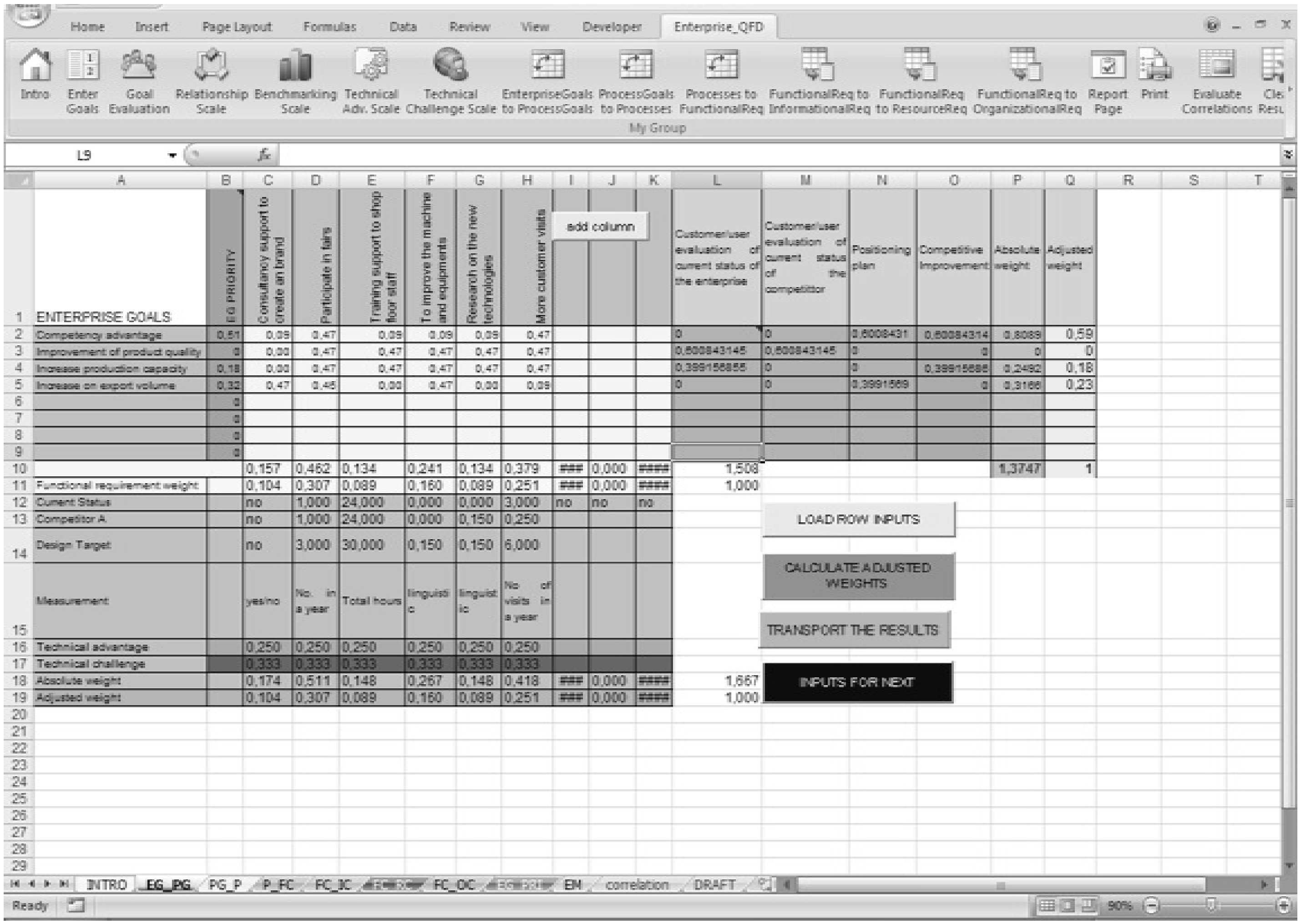


Figure 6.9 Transformation matrix evaluating process goals through enterprise goals.

For the further analysis, correlations can be evaluated among particular characteristics by defining the strength and the sign of the relationship (positive or negative), within the process goals, processes, functional characteristics, informational characteristics, resource characteristics, and organizational characteristics. This phase corresponds to the roof of the QFD matrix where the correlative relations are searched among each pair of expression within only one design characteristic. The matrices are calculated for the transformation of one characteristic into another. During the evaluations, columns are compared with rows, and then quantitative values are assigned to represent the relationship level. Therefore, if any change in a design characteristic exists, then the effects of this change can be seen via these values. However, there can be an additional correlation in each couple of expression within the same design characteristic, and the change may also occur on another one. Thus, the corresponding correlation may also be necessary to display.

Considering this necessity, a worksheet has been added to perform categorical correlation analysis to see whether a positive or negative relationship exists. Therefore, the correlation records can be used to see the effects of any change on the other characteristics, and conflicts can be revealed. Figure 6.10 represents the worksheet designed for the correlation analysis. This phase corresponds to the roof of the matrices in QFD terminology.

The Enterprise-QFD toolbox performs all calculations and analyses about user requirements through different user forms and matrices placed in the worksheets. Finally, the importance values are obtained for enterprise goals, process goals, processes, and further enterprise modelling characteristics, i.e., functional, informational, resource, and organizational characteristics. These findings would be the input values for an enterprise reference architecture selected for modelling. Therefore, requirement analysis results should be gathered as a whole by developing an additional page where the results are collected from the beginning.

Such a report page is placed within the toolbox as a separate worksheet collecting the results of each calculation and transformation operation. Before the calculations, the critical minimum value for the importance values can be determined by the user, and in the report page, these critical low values are marked with red fonts, when the corresponding button is clicked. Furthermore, the printout of the report page can be taken from the results page through the buttons designed within the worksheet. Figure 6.11 shows the partial screen of the report page with the command buttons.

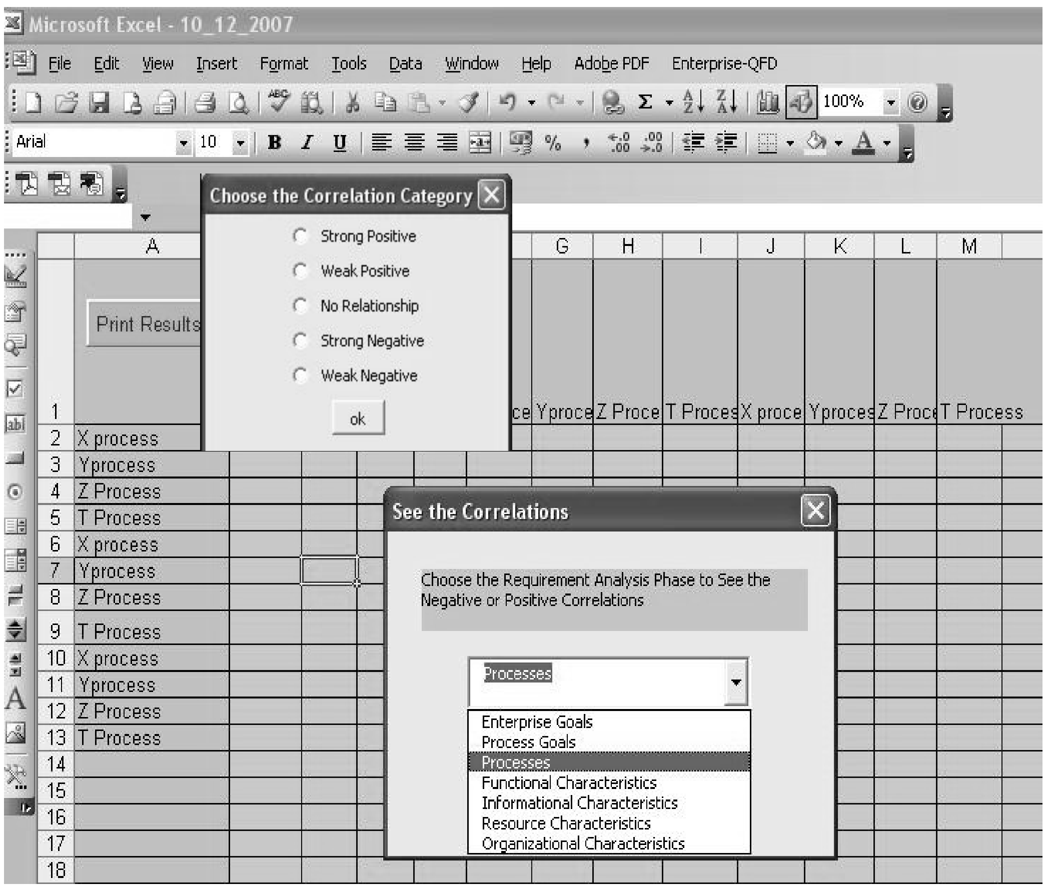
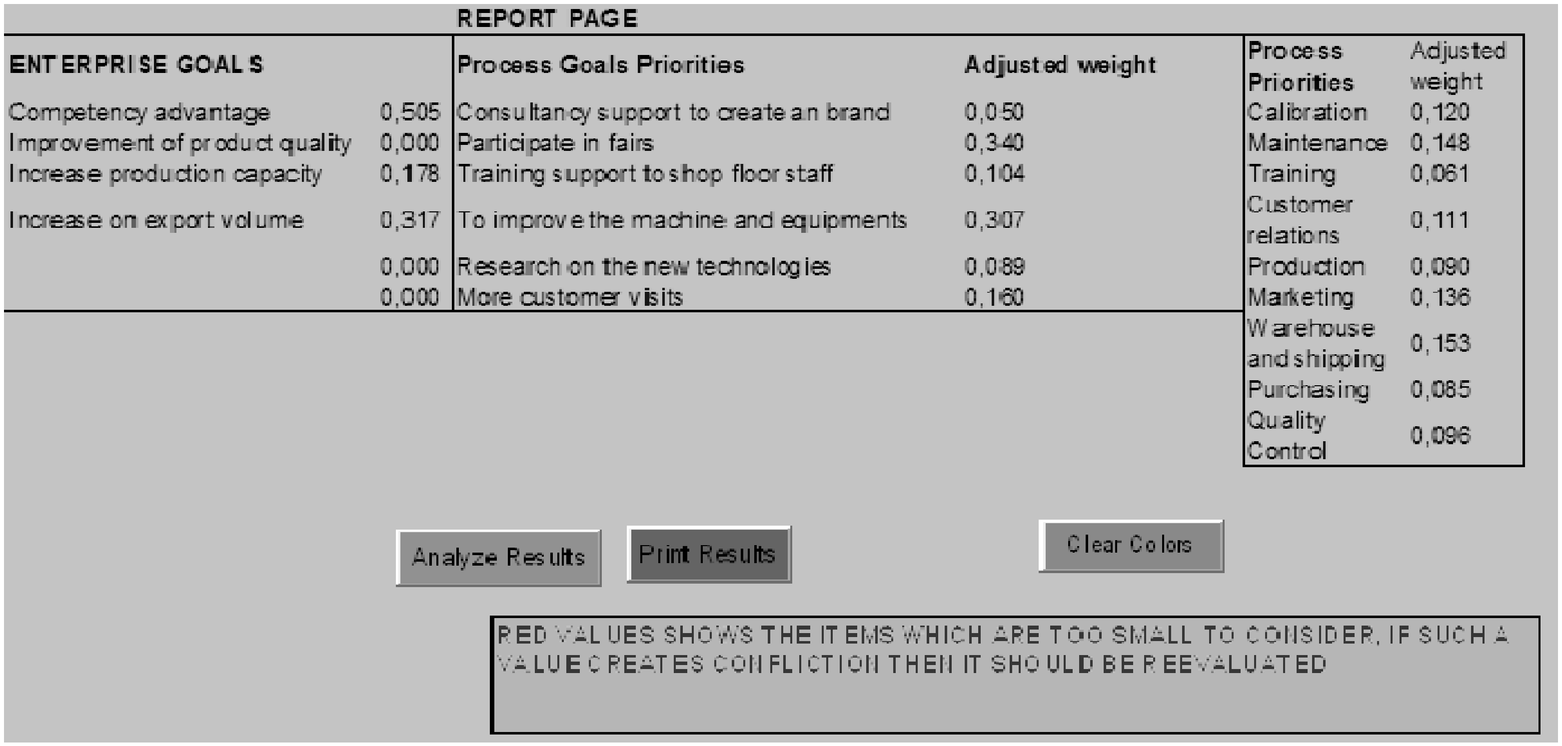


Figure 6.10 Correlation analysis (roof of QFD matrix).



REPORT PAGE

ENTERPRISE GOALS		Process Goals Priorities	Adjusted weight
Competency advantage	0,505	Consultancy support to create an brand	0,050
Improvement of product quality	0,000	Participate in fairs	0,340
Increase production capacity	0,178	Training support to shop floor staff	0,104
Increase on export volume	0,317	To improve the machine and equipments	0,307
	0,000	Research on the new technologies	0,089
	0,000	More customer visits	0,160

Process Priorities	Adjusted weight
Calibration	0,120
Maintenance	0,148
Training	0,061
Customer relations	0,111
Production	0,090
Marketing	0,136
Warehouse and shipping	0,153
Purchasing	0,085
Quality Control	0,096

Figure 6.11 Report page in Enterprise-QFD toolbox.

Beside the worksheets, Enterprise-QFD menu bar provides some functions which would be necessary during implementation. For instance, if “see overall results” function is selected, then the toolbox automatically activates the report page to see the incomplete results whenever it is needed. The “Clear results” function is developed to delete all input values and findings from the toolbox and prepares it for a new implementation. The “Print results” function is added to get printouts of any part of any worksheet through a special user form providing the selection of a particular region.

In summary, this toolbox has been developed to apply Enterprise–QFD approach easily and efficiently. MS Excel VBA modules provide flexibility and ease of use for both programming platform and adoption. In addition to that, the toolbox can be improved according to new requirements in the future by adding new modules and user forms. Another advantage of this toolbox is that it does not require expert knowledge about quantitative methods used in Enterprise-QFD. Thus, the responsible employee can easily make all evaluations following the sequential steps in the menu with a basic instruction of Enterprise-QFD.

6.5 The Transition from Enterprise-QFD to CIMOSA Requirements Modelling

One of the most complex and difficult type of enterprise modelling is to model an enterprise in which computer integrated manufacturing components are implemented. The complexity is due to the need for a complete integration. All relationships between process goals and processes, functional characteristics and processes, functional characteristics and the other modelling constructs, i.e., resources, information, and the organization should be analyzed and clearly defined. The most common reference model in this field is CIMOSA, which is used in enterprises managing computer integrated manufacturing. Enterprise-QFD, as a requirement analysis approach proposed especially for enterprise engineers, can manage all requirement analysis and definition phase of enterprise modelling. Enterprise-QFD supports all modelling components of CIMOSA within the series of the detailed matrices. Matrix in Table 6.12 defines relationships and importance levels of processes and functional characteristics of the enterprise model which corresponds to the functional view of CIMOSA. Then, these functional characteristics are evaluated with respect to the informational, resource, and organizational characteristics, respectively given in Table 6.13 to 6.15. These characteristics also correspond to the related views of CIMOSA, which is one of the most complex reference architectures. The ease of use of the proposed approach is approved during the transformation process from requirements analysis results into CIMOSA constructs. The findings in last four matrices are clarified and ready to be used in the enterprise reference architecture. The characteristics in the matrices are well matched with the reference components during the transformation process. Thus, the outputs of the requirement analysis are directly the inputs of the enterprise reference architectures. Section 6.5 covers examples representing CIMOSA requirement modelling schemes which is developed based on Enterprise-QFD.

The CIMOSA cube that is presented in Figure 3.1 of section 3.3 consists of two main parts in the front side of the cube; the reference architecture and particular architecture. The particular architecture is a set of models documenting the CIM environment of the business users in the process of building their own particular architecture. The CIMOSA cube establishes three modelling levels; requirements definition, design specification, and implementation.

One of the other sides of this cube (See Figure 3.5) is developed based on three generic levels; generic, partial, and particular layer. These layers indicate the level of detail included in the models. The generic layer acts as a library of basic building blocks for constraints, rules, functions, and protocols. The partial layer has predefined partial models which can be applied to particular models. The particular layer is a model of a specific enterprise which is built from the basic blocks and partial models. The last side of the CIMOSA cube represents the modelling views; function, information, resource, and organization. These views are used for modelling manufacturing enterprises.

Enterprise-QFD has been developed by considering that the closer the outputs of requirements analysis to the requirement modelling component of the reference architecture are the easier to transform the results of the analysis into the reference architecture is.

Enterprise-QFD deploys the process information into functional objects by transforming the relationships and importance weights of the processes after the deployment of the process goals. In the next step, the information on the functional entities are deployed into informational, organizational, and resource entities, respectively. These series of transformation is carried out by considering a particular architecture. The use of Enterprise-QFD can be represented on the CIMOSA. Thus, the deployment process is handled with respect to the views of CIMOSA (function, information, organization, and resource). If the reference architecture selected for modelling is different from these views, then the matrices can be constructed upon the selected reference views.

During the calculation of importance weights of the entities, Enterprise-QFD matrices take the relationships between the expressions in the rows and columns, and other planning issues integrated within the matrices. These relationships support the requirements model of the enterprise. If one of the matrices is considered (see Table 6.11) deploying the process goals into processes, it can be seen that the relationship places in the middle of the matrix indicates the relationships between each process goal and process. Therefore, blank processes or process goals can be revealed, if no relationship is assigned. Furthermore, if the roof of the matrix is applied to figure out the correlations between each couple of processes, positively or negatively, then these relationships can be used in the initial phase of the enterprise model including the domains, domain processes, and domain relationships.

The outcomes of Enterprise-QFD are applied to the CIMOSA requirements modelling phase considering the views in the CIMOSA cube (see Figure 3.5), and sample schemes are developed including a domain process, domain relationship, resource and organizational unit definition.

The first step to represent the requirements model in an enterprise is to construct the domain structure. A domain structure figures out the domains and their processes covered by the enterprise model and object views used by each couple of domains. In the sample case study, the domain structure is described as in Figure 6.12 including four domains and eleven domain processes. Each pair of domains is tied by the object views. Object views correspond to the information objects or characteristics defined by the findings of Enterprise-QFD in Table 6.15. Some of the object views are used by more than one domains and the frequency of usage indicates the importance of them. The intensity of usages of object views can easily be seen from the importance weights in Table 6.15 where the functional requirements are deployed into the information objects. After the domain structure is described within the enterprise, each domain is defined in details.

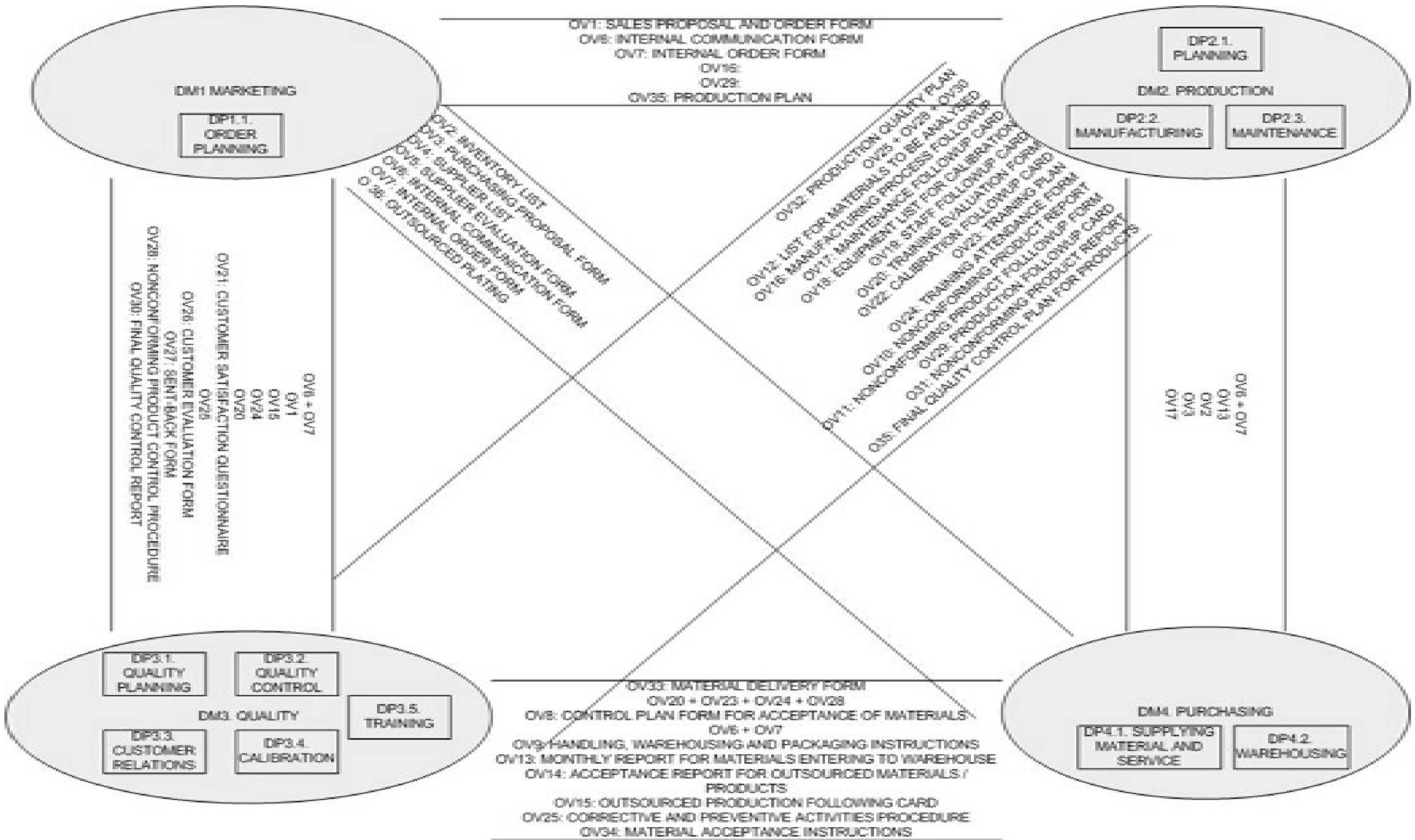


Figure 6.12 Domain structure of the enterprise.

Domains defined in the model can be represented by different schemes. Each scheme differentiates with respect to the details and target of use. The generic representations of the domains are figured out by the domain definitions. Figure 6.13 to Figure 6.16 are the domain definitions showing the components of Marketing Domain (DM1) Production Domain (DM2), Quality Domain (DM3), and Purchasing Domain (DM4), respectively.

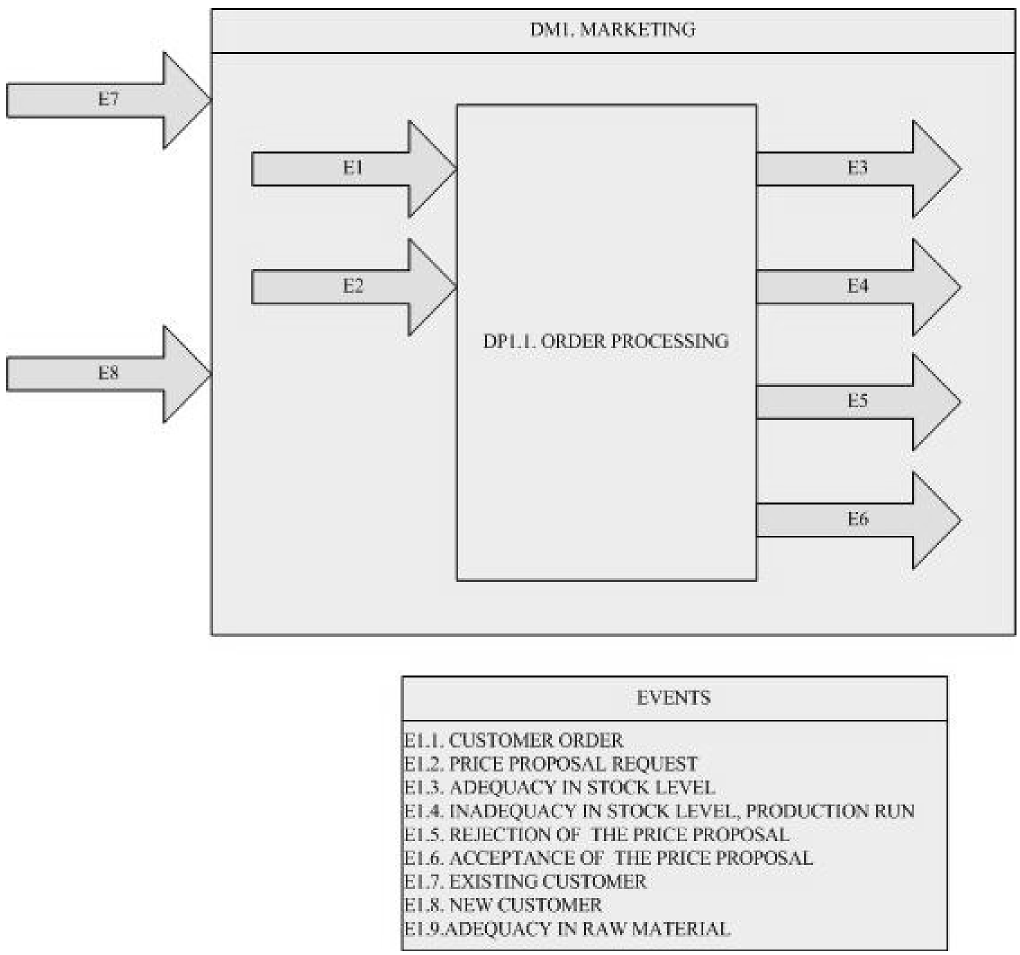


Figure 6.13 Domain definition of Marketing (DM1).

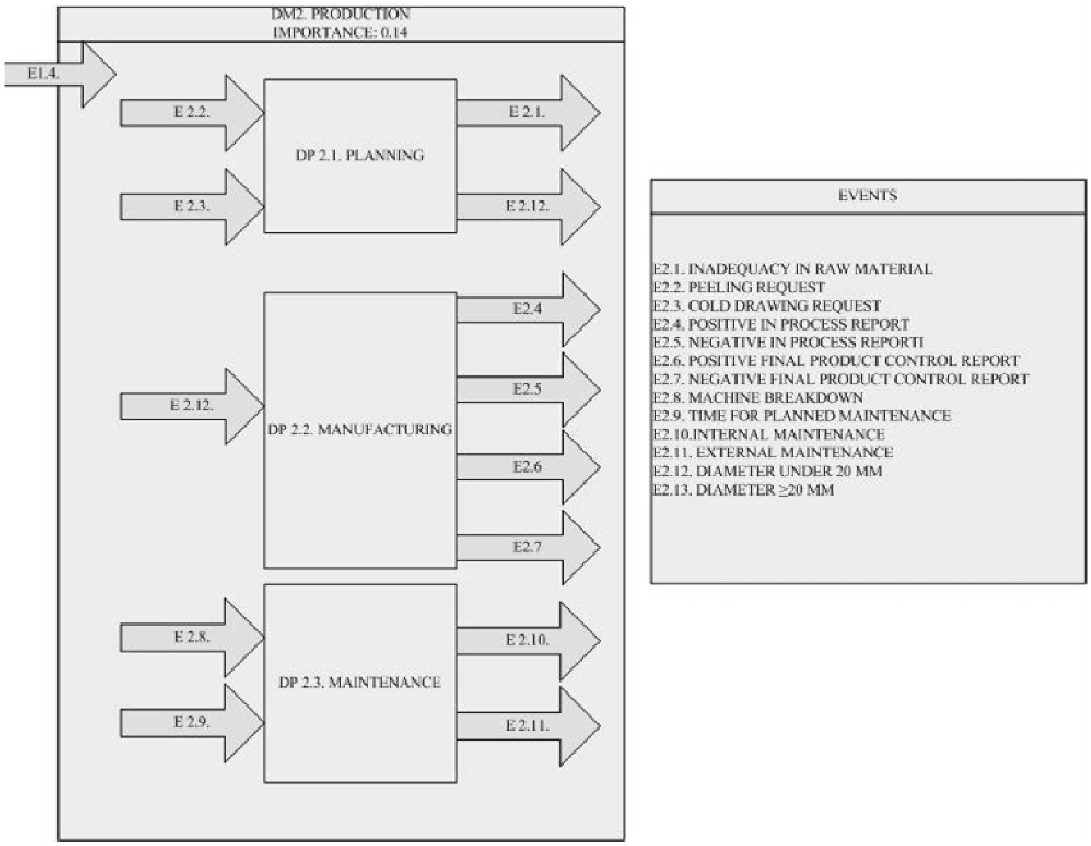


Figure 6.14 Domain definition of Production(DM2).

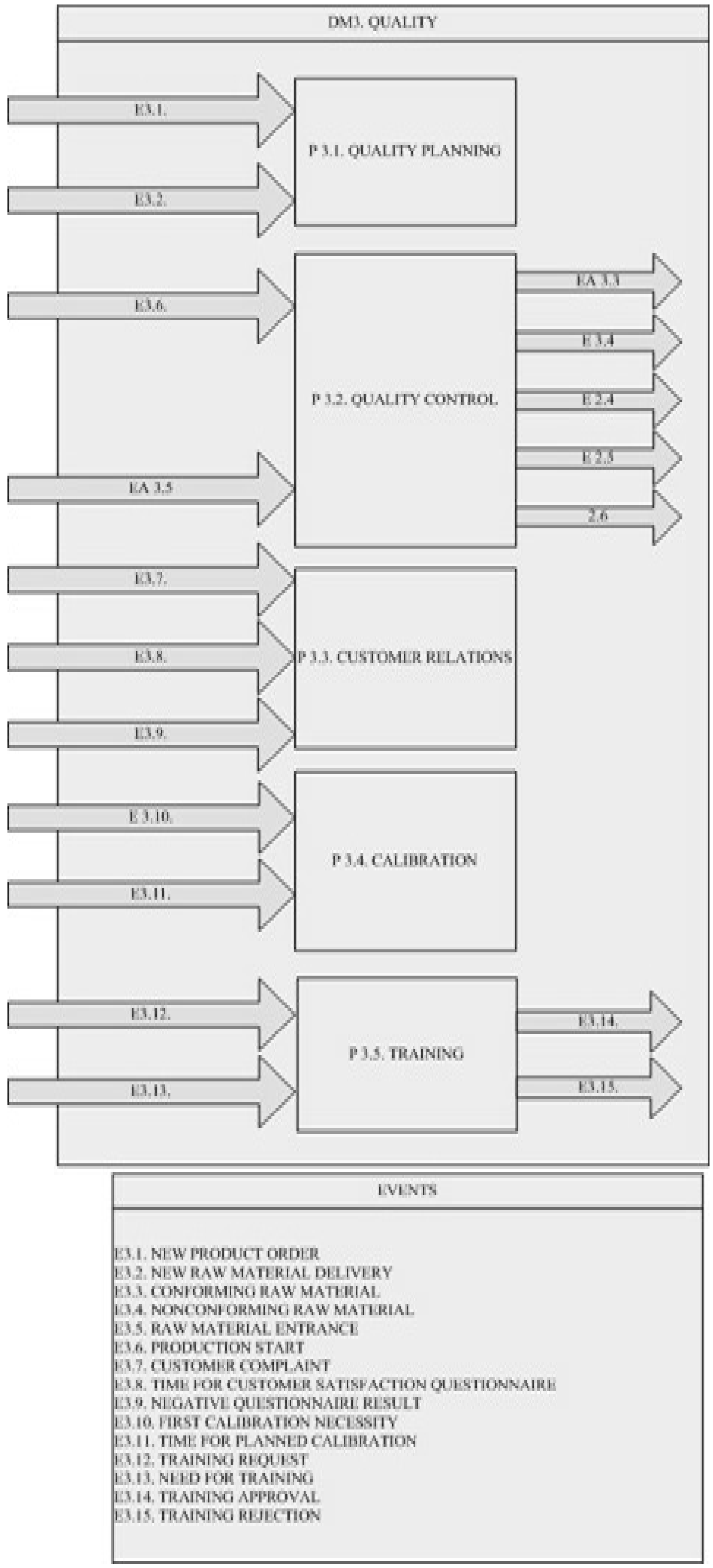


Figure 6.15 Domain definition of Quality (DM3).

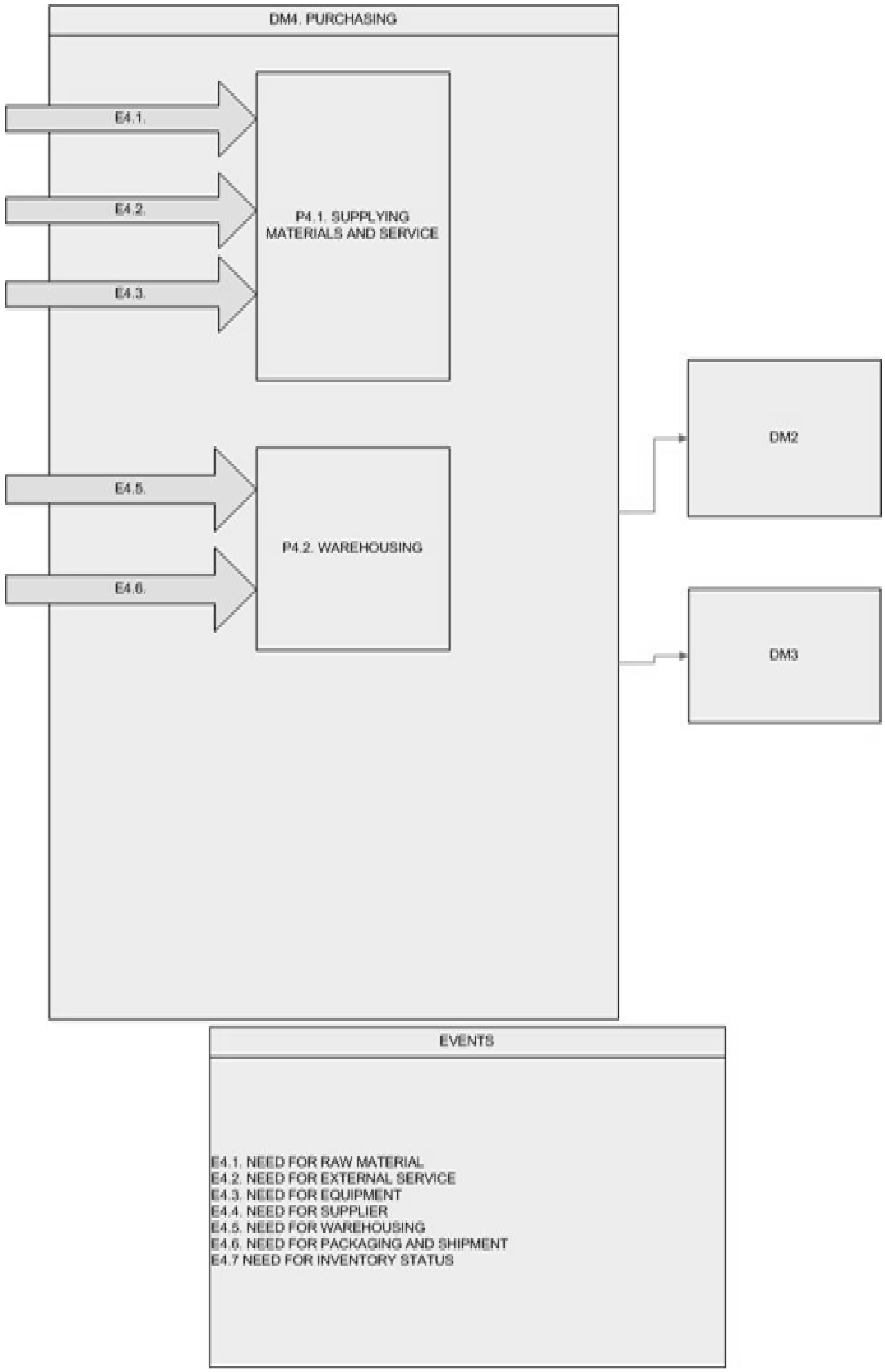


Figure 6.16 Domain definition of Purchasing (DM4).

Domain definitions from Figure 6.13 to 6.16 represent the boundaries of the domains, its domain processes triggering the events and the processes to start the domain, and the output events and domains/processes related to the corresponding domain. For example, the Production Domain in Figure 6.14 includes three domain processes (DP): Planning, Manufacturing, and Maintenance. There exist thirteen triggering events related to the Production Domain.

Figure 6.14 is the initial and generic definition of the Production Domain representing the domain processes and the events that trigger the activities within the domain processes. Domain processes are gathered and analyzed in Table 6.11 to ensure that process goals cover all processes or each process is covered by at least one process goal. Table 6.11 provides the required information for domain definitions, e.g. the Production Domain in Figure 6.14. The importance values or importance weights are obtained just after the evaluations are completed for the relationships between each process goal and process pair. The total importance weight of Production Domain is 0.19 including the Warehousing Process. The production process is broken down into two processes after analyzing the functional characteristics in Table 6.12.

Another type of domain definition is carried out through definition statements. Figure 6.17 shows a definition statement for Marketing domain that covers the domain objectives, domain processes, events (E), constraints, and object views (OV). In the domain statements, identifier abbreviation and the name of the domain are defined at the top of the statement card. With the use of Enterprise-QFD, the importance value of the domain can also be added from Table 6.11 to this field of the statement card. In the next part of the card, the objectives, constraints, and processes of the domain are defined and functional entities (FE) are added as the remaining boundary of the domain. After these definitions, events and object views are listed related with the domain.

DOMAIN

IDENTIFIER:DM1
NAME: MARKETING
IMPORTANCE: 0.21

DOMAIN OBJECTIVES: TO MANAGE THE CUSTOMER ORDERS, IMPROVE THEIR SATISFACTION, TO FIND NEW CUSTOMERS, AND CONTROL ALL ACTIVITIES WITHIN THE COMPANY TO MEET THE EXPECTATIONS OF CUSTOMERS.

DOMAIN CONSTRAINTS: PRODUCTION CAPACITY, SALESMAN CAPACITY
DOMAIN PROCESSES: DP1.1/ORDER PROCESSING
BOUNDARY: FE 1 TO 9

EVENTS:
E1.1/CUSTOMER ORDER
E1.2/PRICE PROPOSAL REQUEST
E1.3/ADEQUACY IN STOCK LEVEL
E1.4/INADEQUACY IN STOCK LEVEL, PRODUCTION RUN
E1.5/REJECTION OF THE PRICE PROPOSAL
E1.6/ACCEPTANCE OF THE PRICE PROPOSAL
E1.7/EXISTING CUSTOMER
E1.8/NEW CUSTOMER
E1.9/ADEQUACY IN RAW MATERIAL

OBJECT VIEWS:
OV1/SALES PROPOSAL AND ORDER FORM
OV6/INTERNAL COMMUNICATION FORM
OV7/INTERNAL ORDER FORM
OV15/OUTSOURCED PRODUCTION FOLLOWUPCARD
OV21/CUSTOMER SATISFACTION QUESTIONNAIRE
OV26/CUSTOMER EVALUATION FORM
OV27/SEND-BACK FORM

Figure 6.17 Statement type domain definition example for Marketing (DM1).

Similar definitions are also carried out while the relationships within each couple of domains. Domain relationship cards are the detailed arrangements of the domain structure given in Figure 6.12. A domain relationship includes common object views, events involved by the domains, importance values of object views, and their frequency of occurrences. Figure 6.18 presents the relationships between Marketing Domain and other Domains; Production and Quality Domains, respectively. Figure 6.19 describes the relationships between Purchasing-Quality and Purchasing-Production Domains.

DOMAIN RELATIONSHIP
IDENTIFIER:DR1
NAME: ORDERPROC&PRODUCTION

DOMAIN1 NAME: DM1/MARKETING (0.21)
DOMAIN2 NAME: DM2/PRODUCTION (0.14)
INVOLVED OBJECT VIEWS:
IDENTIFIER: OV1 (0.02)
NAME: SALES PROPOSAL AND ORDER FORM
FROM: DM1
TO: DM2
IDENTIFIER: OV6 (0.04)
NAME: INTERNAL COMMUNICATION FORM
FROM:DM1- DM2
TO: DM2-DM1
IDENTIFIER: OV7
NAME: INTERNAL ORDER FORM (0.05)
FROM: DM2-DM1
TO:DM1-DM2
IDENTIFIER: OV16 (0.06)
NAME: MANUFACTURING FOLLOWUP
FROM:DM2
TO: DM1

INVOLVED EVENTS:
IDENTIFIER: E1.4
NAME: INADEQUACY IN STOCK
LEVEL(PRODUCT)
FROM: DM1
TO: DM2
FREQUENCY: 4 PER DAY

DOMAIN RELATIONSHIP
IDENTIFIER:DR3
NAME: ORDERPROC&PURCHASING

DOMAIN1 NAME: DM1/MARKETING (0.21)
DOMAIN2 NAME: DM4/PURCHASING (0.09)
INVOLVED OBJECT VIEWS:
IDENTIFIER: OV2 (0.02)
NAME: INVENTORY LIST
FROM:DM4
TO: DM1
IDENTIFIER: OV6 (0.04)
NAME: INTERNAL COMMUNICATION FORM
FROM:DM1-DM4
TO: DM4-DM1
IDENTIFIER: OV7 (0.05)
NAME: INTERNAL ORDER FORM
FROM: DM1
TO: DM4
IDENTIFIER:

INVOLVED EVENTS:
IDENTIFIER: E4.7
NAME: NEED FOR INVENTORY STATUS
FROM: DM1
TO: DM4
FREQUENCY: 4 PER DAY
IDENTIFIER: E4.6
NAME: NEED FOR PACKAGING & SHIPMENT
FROM: DM1-DM4
TO: DM4-DM1
FREQUENCY: 2 PER DAY

DOMAIN RELATIONSHIP
IDENTIFIER:DR2
NAME: ORDERPROC&QUALITY

DOMAIN1 NAME: DM1/MARKETING (0.21)
DOMAIN2 NAME: DM3/QUALITY (0.13)
INVOLVED OBJECT VIEWS:
IDENTIFIER: OV6 (0.04)
NAME: INTERNAL COMMUNICATION FORM
FROM: DM3
TO: DM1
IDENTIFIER: OV7 (0.05)
NAME: INTERNAL ORDER FORM
FROM: DM1
TO: DM3
IDENTIFIER: OV15
NAME: OUTSOURCED PRODUCTION FOLLOWUP FORM
(0.03)
FROM: DM1
TO: DM3
IDENTIFIER: OV24 (0.01)
NAME: TRAINING ATTENDANCE FORM
FROM: DM1
TO: DM3
IDENTIFIER: OV20 (0.01)
NAME: TRAINING EVALUATION FORM
FROM: DM1
TO: DM3
IDENTIFIER: OV21 (0.01)
NAME: CUSTOMER SATISFACTION QUESTIONNAIRE
FROM: DM1-DM3
TO: DM3-DM1
IDENTIFIER: OV25 (0.03)
NAME: CORRECTIVE AND PREVENTIVE ACTIVITIES
PROCDURE
FROM: DM3
TO: DM1
IDENTIFIER: OV26 (0.02)
NAME: CUSTOMER EVALUATION FORM
FROM: DM1-DM3
TO: DM3-DM1
IDENTIFIER: OV20
NAME: TRAINING EVALUATION FORM (0.01)
FROM: DM1
TO: DM3
IDENTIFIER: OV27
NAME: SEND BACK FORM (0.02)
FROM: DM1
TO: DM3
IDENTIFIER: OV28 (0.03)
NAME: NONCONFORMING PRODUCT PROCEDURE
FROM: DM3
TO: DM1
IDENTIFIER: OV23 (0.04)
NAME: FINAL QUALITY CONTROL REPORT
FROM: DM3
TO: DM1
IDENTIFIER: OV23 (0.01)
NAME: TRAINING PLAN
FROM: DM3
TO: DM1

INVOLVED EVENTS:
IDENTIFIER: E3.1
NAME: NEW PRODUCT REQUEST
FROM: DM1
TO: DM3
FREQUENCY: 3 PER MONTH
IDENTIFIER: E3.7
NAME: CUSTOMER COMPLAINT
FROM:DM1
TO: DM3
FREQUENCY: 1 PER MONTH
IDENTIFIER: E3.9
NAME: NEGATIVE RESULTS IN CUSTOMER
QUESTIONNAIRE
FROM: DM3
TO:DM1
FREQUENCY: 1-2 PER YEAR
IDENTIFIER: E3.12
NAME:TRAINING REQUEST
FROM:DM4
TO: DM3
FREQUENCY: 2 PER YEAR
IDENTIFIER: E3.13
NAME: NEED FOR TRAINING
FROM:DM1
TO: DM3
FREQUENCY: 2 PER YEAR

Figure 6.18 Examples for domain relationships-1(Order Processing-Purchasing, Order Processing-Production, Order Processing-Quality).

DOMAIN RELATIONSHIP
IDENTIFIER:DR6
NAME: QUALITY ACTIVITIES&PURCHASING
DOMAIN1 NAME: DM3/QUALITY
DOMAIN2 NAME: DM4/PURCHASING

INVOLVED OBJECT VIEWS:
IDENTIFIER: OV6 (0.04)
NAME: INTERNAL COMMUNICATION FORM
FROM: DM3-DM4
TO: DM4-DM3
IDENTIFIER: OV7 (0.05)
NAME: INTERNAL ORDER FORM
FROM: DM4-DM3
TO: DM3-DM4
IDENTIFIER: OV23 (0.01)
NAME: TRAINING PLAN
FROM: DM3
TO: DM4
IDENTIFIER: OV24 (0.01)
NAME: TRAINING ATTENDANCE FORM
FROM: DM4
TO: DM3
IDENTIFIER: OV20 (0.02)
NAME: TRAINING EVALUATION FORM
FROM: DM4
TO: DM3
IDENTIFIER: OV25 (0.03)
NAME: CORRECTIVE AND PREVENTIVE ACTIVITIES PROCEDURE
FROM: DM3
TO: DM4
IDENTIFIER: OV13 (0.02)
NAME: MONTHLY REPORT FOR MATERIALS ACCEPTED BY WAREHOUSE
FROM: DM4
TO: DM3
IDENTIFIER: OV14 (0.02)
NAME:ACCEPTANCE REPORT FOR OUTSOURCED PRODUCTS
FROM: DM4
TO: DM3
IDENTIFIER: OV34 (0.03)
NAME: MATERIAL ACCEPTANCE INSTRUCTIONS
FROM: DM3
TO: DM4

IDENTIFIER: OV8 (0.02)
NAME: CONTROL PLAN FOR ACCEPTANCE OF MATERIALS
FROM: DM3
TO: DM4
IDENTIFIER: OV28 (0.02)
NAME: NONCONFORMING PRODUCT PROCEDURE
FROM: DM3
TO: DM4
IDENTIFIER: OV33 (0.03)
NAME: MATERIAL DELIVERY FORM
FROM: DM3
TO: DM4
IDENTIFIER: OV9 (0.03)
NAME: HANDLING, WAREHOUSING & PACKAGING INSTRUCTIONS
FROM: DM3
TO: DM4
IDENTIFIER: OV22 (0.02)
NAME: CALIBRATION FOLLOWUP CARD
FROM: DM3
TO: DM2
INVOLVED EVENTS:
IDENTIFIER: E3.5.
NAME:RAW MATERIAL DELIVERY
FROM: DM4
TO: DM3
FREQUENCY: 1 PER DAY
IDENTIFIER: E3.2
NAME: NEW RAW MATERIAL DELIVERY
FROM: DM4
TO: DM3
FREQUENCY: 1-2 PER YEAR
IDENTIFIER: E3.3
NAME: ACCEPTABLE RAW MATERIAL
FROM: DM3
TO: DM4
FREQUENCY: 1 PER DAY
IDENTIFIER: E3.4
NAME: UNACCEPTABE RAW MATERIAL
FROM: DM3
TO: DM4
FREQUENCY: 3 PER YEAR
IDENTIFIER: E3.13
NAME: NEED FOR TRAINING
FROM:DM4
TO: DM3
FREQUENCY: 2 PER YEAR
IDENTIFIER: E3.12
NAME:TRAINING REQUEST
FROM:DM4
TO: DM3
FREQUENCY: 2 PER YEAR

DOMAIN RELATIONSHIP
IDENTIFIER:DR5
NAME: PRODUCTION&PURCHASING

DOMAIN1 NAME: DM2/PRODUCTION (0.14)
DOMAIN2 NAME: DM4/PURCHASING (0.09)

INVOLVED OBJECT VIEWS:
IDENTIFIER: OV6 (0.04)
NAME: INTERNAL COMMUNICATION FORM
FROM: DM3-DM4
TO: DM4-DM3
IDENTIFIER: OV7 (0.05)
NAME: INTERNAL ORDER FORM
FROM: DM4-DM3
TO: DM3-DM4
IDENTIFIER: OV2 (0.02)
NAME: INVENTORY LIST
FROM:DM4
TO: DM2
IDENTIFIER: OV13 (0.02)
NAME: MONTHLY REPORT FOR MATERIALS ACCEPTED BY WAREHOUSE
FROM: DM4
TO: DM2
IDENTIFIER: OV17 (0.05)
NAME: MAINTENANCE FOLLOWUP
FROM: DM2
TO: DM4

INVOLVED EVENTS:
IDENTIFIER: E4.1
NAME:NEED FOR RAW MATERIAL
FROM: DM2
TO: DM4
FREQUENCY: 1 PER WEEK
IDENTIFIER: E2.1
NAME:INADEQUACY IN RAW MATERIAL/ PRODUCT
FROM: DM4
TO: DM2
FREQUENCY: 1-2 PER WEEK

Figure 6.19 Domain relationships-2
(Purchasing –Quality), (Purchasing –Production).

The domain relationships can easily be seen from Tables 6.11 to 6.15 including functional, informational, organizational, and resource characteristics. Each domain includes at least one domain process which carries out the functional tasks within the domain. Domain Processes are indexed as the domain definitions and represented graphically with the related events and other processes. Figure 6.20 to 6.30 shows the domain process definitions with the input and output characteristics indicating the related triggering events and processes.

Figure 6.22 is one of the example schemes for graphical representation of a domain process. Manufacturing Domain Process (DP 2.2) is represented according to the operations flow and events. The information to represent this domain process comes from the functional entities given in Table 6.12. Each enterprise activity has its own importance value, e.g. grinding (0.03). Figure 6.22 also indicates the other domain processes related with Manufacturing such as Warehousing and Quality Control. These relationships are first represented in Table 6.12.

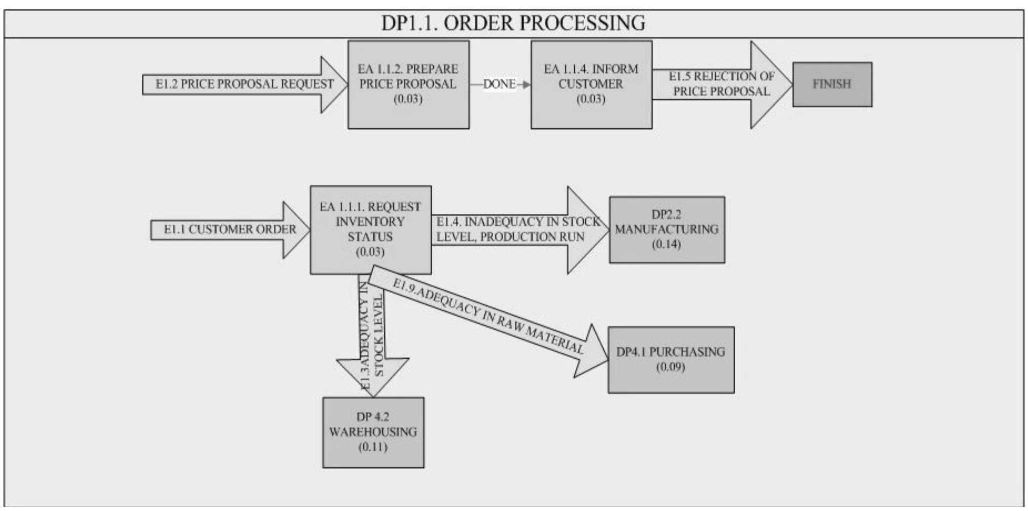


Figure 6.20 Order processing (DP 1.1).

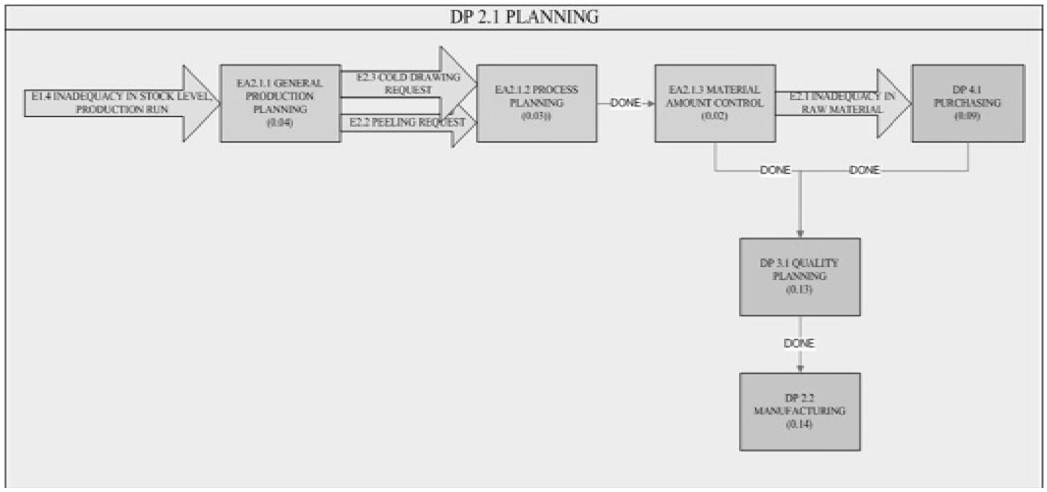


Figure 6.21 Planning process (DP 2.1).

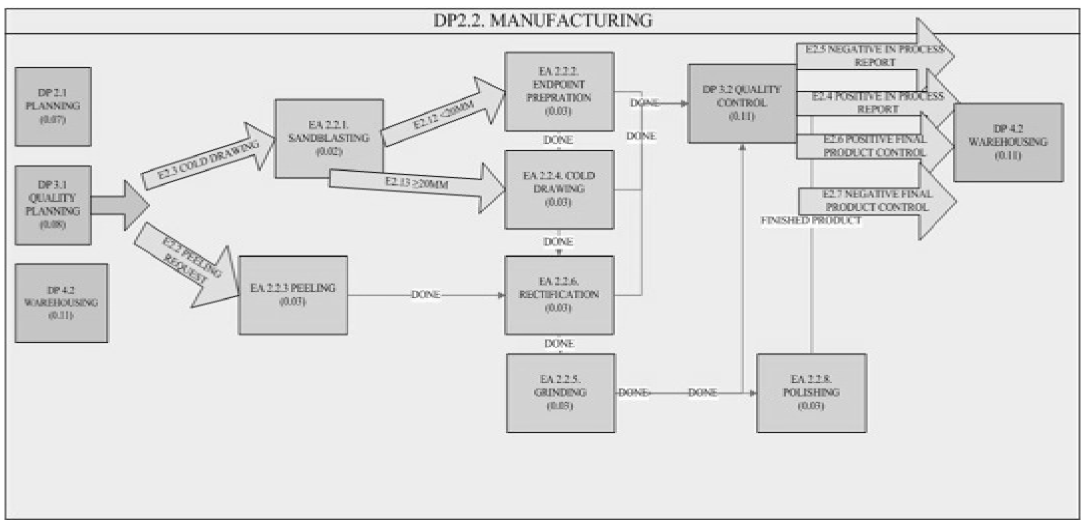


Figure 6.22 Manufacturing process (DP 2.2).

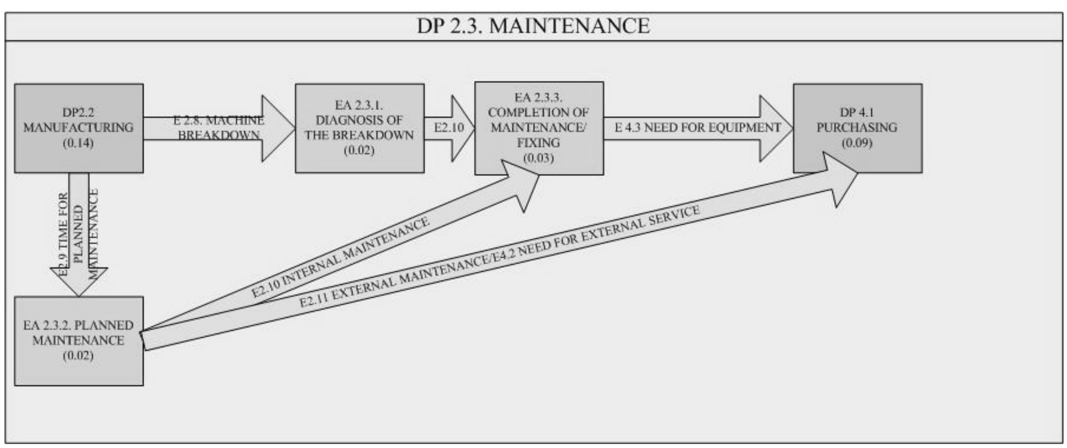


Figure 6.23 Maintenance process (DP 2.3).

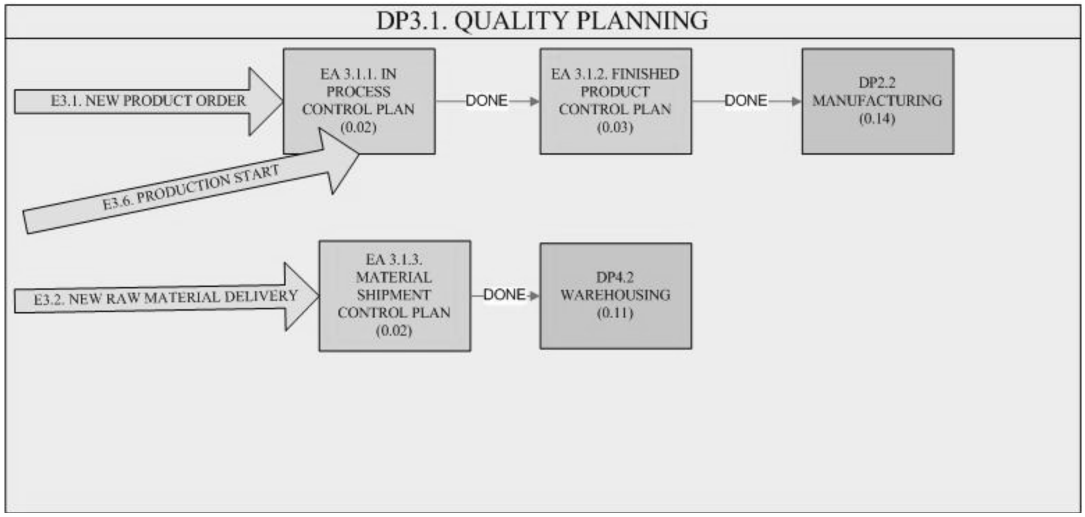


Figure 6.24 Quality Planning process (DP 3.1).

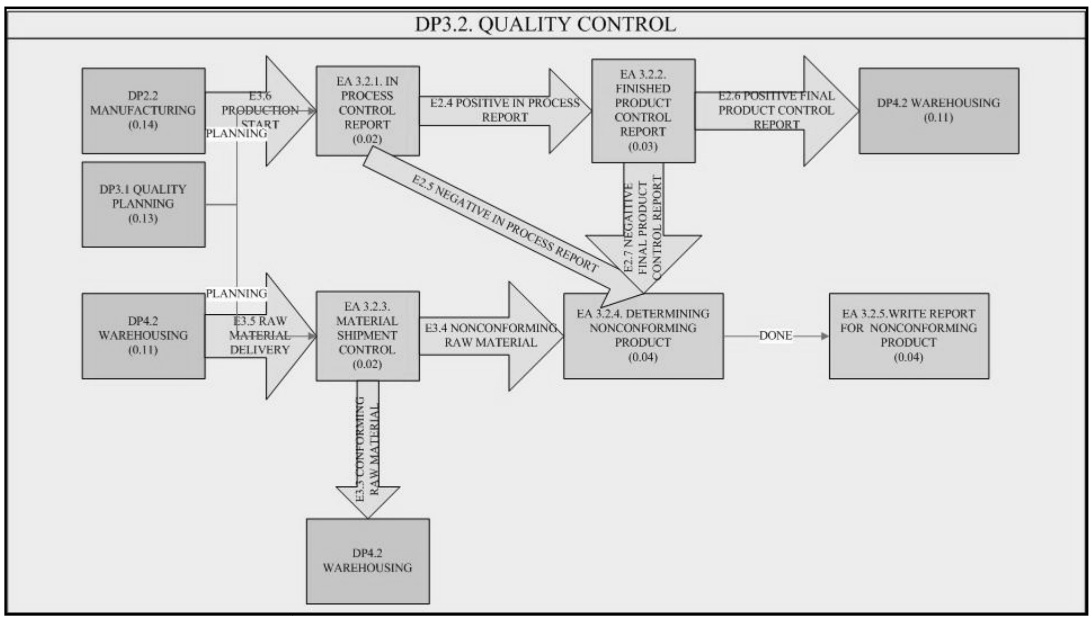


Figure 6.25 Quality Control process (DP 3.2).

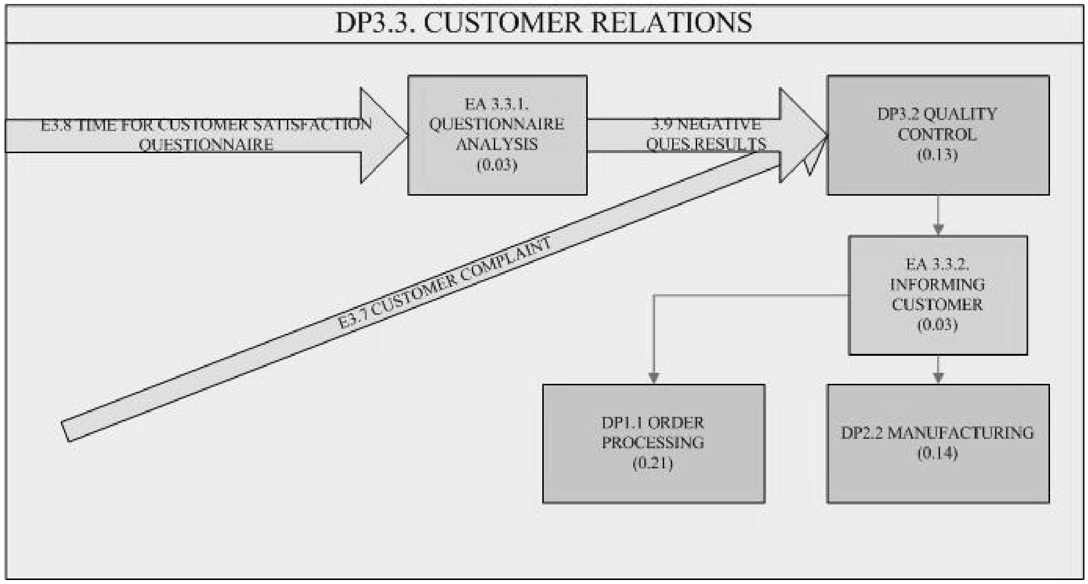


Figure 6.26 Customer Relations process (DP 3.3).

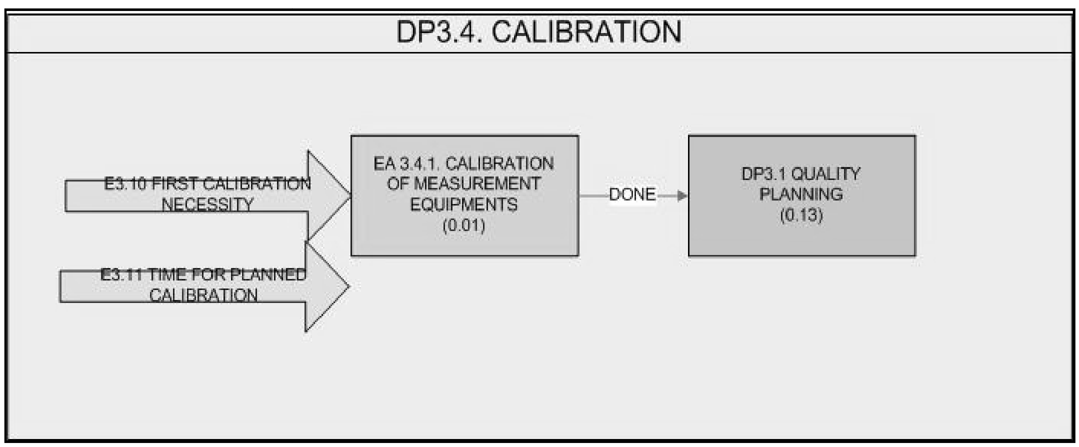


Figure 6.27 Calibration process (DP 3.4).

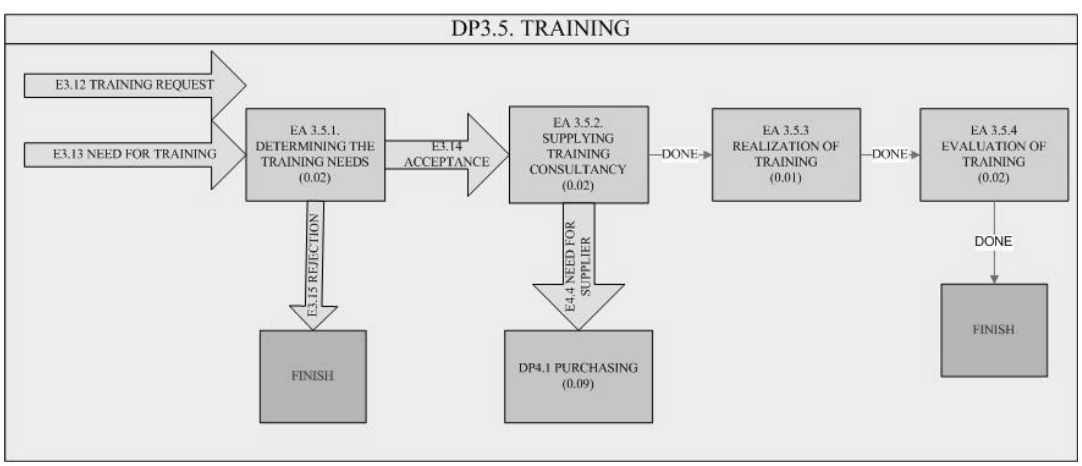


Figure 6.28 Training process (DP 3.5).

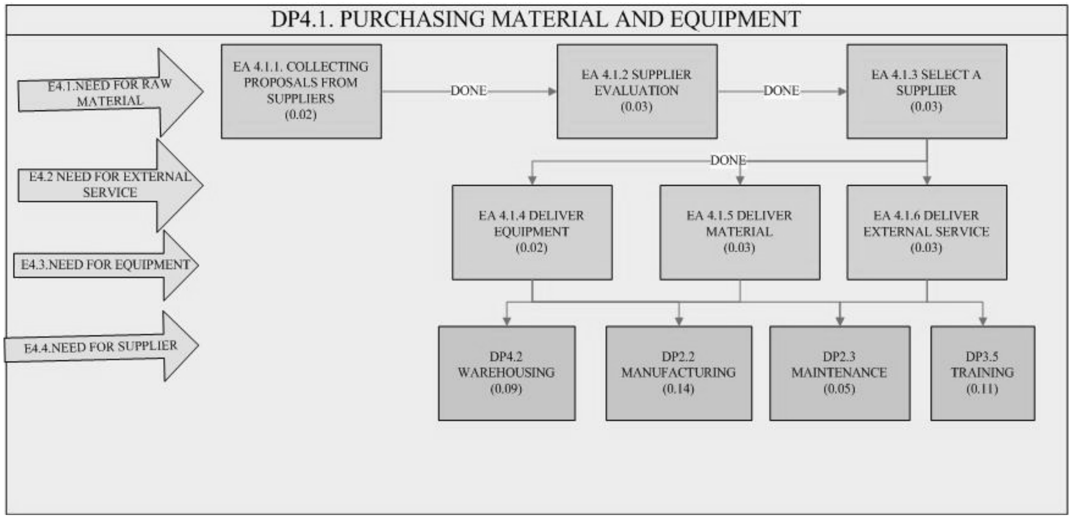


Figure 6.29 Purchasing process (DP 4.1).

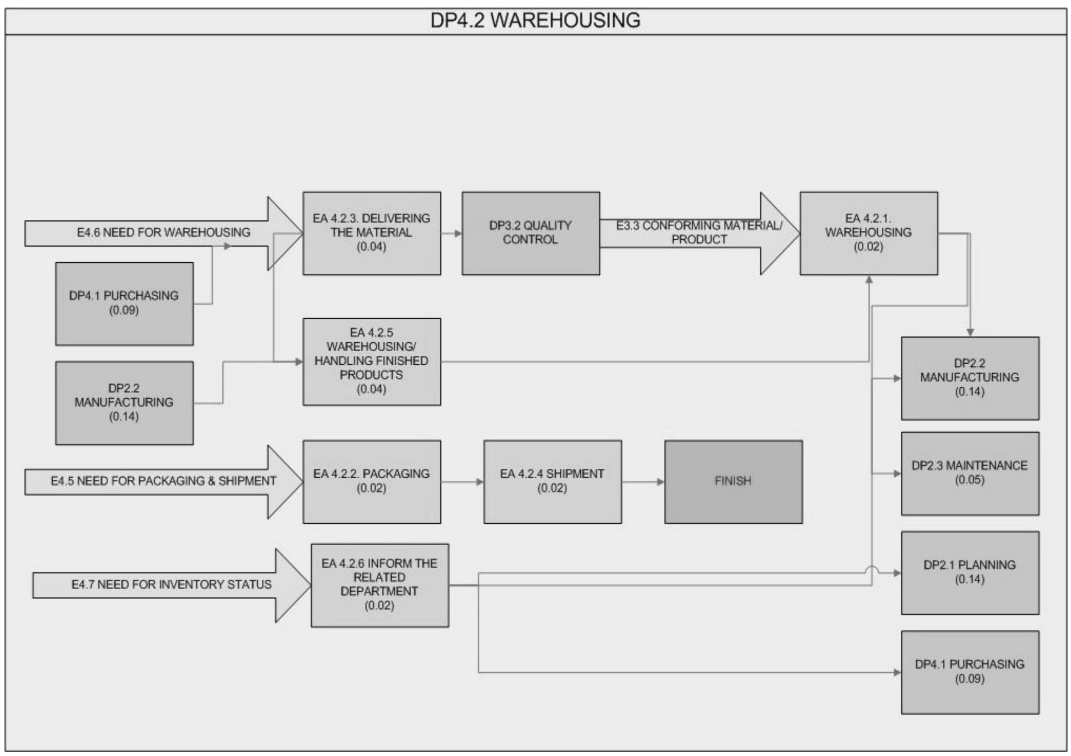


Figure 6.30 Warehousing process (DP 4.2).

The requirements modelling phase in CIMOSA also provides behavioural analysis of the processes. For this purpose, a behavioural rule set is defined for each process. The rule set describes similar definitions as domain definition card including domain objectives, constraints, and related events. Besides, The Domain process card includes detailed description of the process in verbal statements and rules to follow with respect to the sequence of enterprise activities (EA) and their ending status (ES). Behavioural rule set of The Manufacturing Domain Process is presented as an example in Figure 6.31.

DOMAIN PROCESS
IDENTIFIER:DP2.2
NAME: MANUCACTURING

DESCRIPTION: RECEIVES PRODUCTION PLAN FROM DP2.1, QUALITY PLAN FROM DP3.1, AND REQUIRED MATERIALS FROM DP4.1 AND DP4.2, THEN PERFORMS MANUFACTURING OPERATIONS TO FINISH THE PRODUCT.
DOMAIN OBJECTIVES: TO EXECUTE THE PRODUCTION PLAN AND OBTAIN HIGH QUALTY PRODUCTS ON TIME
CONSTRAINTS: PRODUCTION CAPACITY,
DECLARATIVE RULES: FOLLOW THE OPERATION SEQUENCE.

EVENTS: E2.2, E2.3, E2.4, E2.5, E2.6, E2.7, E2.12, E2.13
PROCESS BEHAVIOR:
WHEN (START WITH PRODUCTION PLAN EXECUTION) DO MANUFACTURING
WHEN (ES(PEELING REQUEST)=OK) DO PEELING
WHEN (ES(PEELING)=DONE) THEN DO QUALITY CONTROL
WHEN (ES(QUALITY CONTROL)= OK) DO RECTIFICATION
WHEN (ES(QUALITY CONTROL)= NOT OK) DO FINISH

WHEN (ES(COLD DRAWING REQUEST)=OK) DO SANDBLASTING
WHEN(ES(SANDBLASTING)=DONE) DO QUALITY CONTROL
WHEN (ES(QUALITY CONTROL)= OK) DO COLD DRAWING
WHEN (ES(QUALITY CONTROL)= NOT OK) DO FINISH

WHEN (ES(DIAMATER<20 MM)=YES) DO ENDPOINT PREPARATION
WHEN (ES(ENDPOINT PREPARATION)= DONE) DO QUALITY CONTROL
WHEN (ES(QUALITY CONTROL)= OK) DO COLD DRAWING

WHEN (ES(QUALITY CONTROL)= NOT OK) DO FINISH
WHEN (ES(DIAMETER<20=NO) DO COLD DRAWING
WHEN(ES(COLD DRAWING)=DONE) DO QUALITY CONTROL
WHEN (ES(QUALITY CONTROL)= OK) DO RECTIFICATION
WHEN (ES(QUALITY CONTROL)= NOT OK) DO FINISH

WHEN (ES(RECTIFICATION)=DONE)) DO QUALITY CONTROL
WHEN (ES(QUALITY CONTROL)= OK) DO GRINDING
WHEN (ES(QUALITY CONTROL)= NOT OK) DO FINISH
WHEN (ES(GRINDING)= DONE) DO QUALITY CONTROL
WHEN (ES(QUALITY CONTROL)= OK) DO POLISHING
WHEN (ES(QUALITY CONTROL)= NOT OK) DO FINISH
WHEN(ES(POLISHING)=DONE) DO QUALITY CONTROL (FINISHED PRODUCT)
WHEN (ES(QUALITY CONTROL)= OK) DO WAREHOUSING
WHEN (ES(QUALITY CONTROL)= NOT OK) DO FINISH

COMPRISES: EA2.2.1, EA2.2.2, EA2.2.3, EA2.2.4, EA2.2.5, EA2.2.6, EA2.2.8, DP4.2, DP3.1, DP3.2, DP2.1

Figure 6.31 Behavioural Rules Example (DP2.2).

The definitions so far are closely related with functional characteristics of domains and domain processes. After this stage, additional entities and objects about other views, i.e., information, organization, and resource views are constructed. Figure 6.32 and 6.33 represents samples of resource, information object, and organizational unit from these views. These definitions can be obtained by

Enterprise-QFD Matrices. For instance, Object View (OV) of Production Follow up Card is gathered from Table 6.13 with the importance weight of 0.06; Production Operator as an organization unit with the importance weight of 0.15 is obtained from Table 6.14; and Sand Blasting Machine is an entity for resource view (Functional entity: FE) with the importance weight of 0.17 as given in Table 6.15 and defined in Figure 6.32. These definitions and importance weights are analyzed with respect to the relationship intensity with all functional characteristics within the enterprise. Enterprise Activities (EA) are also defined in CIMOSA, e.g. Sandblasting activity is defined in details in Figure 6.32 and 6.33 and this activity is also analyzed in Table 6.12.

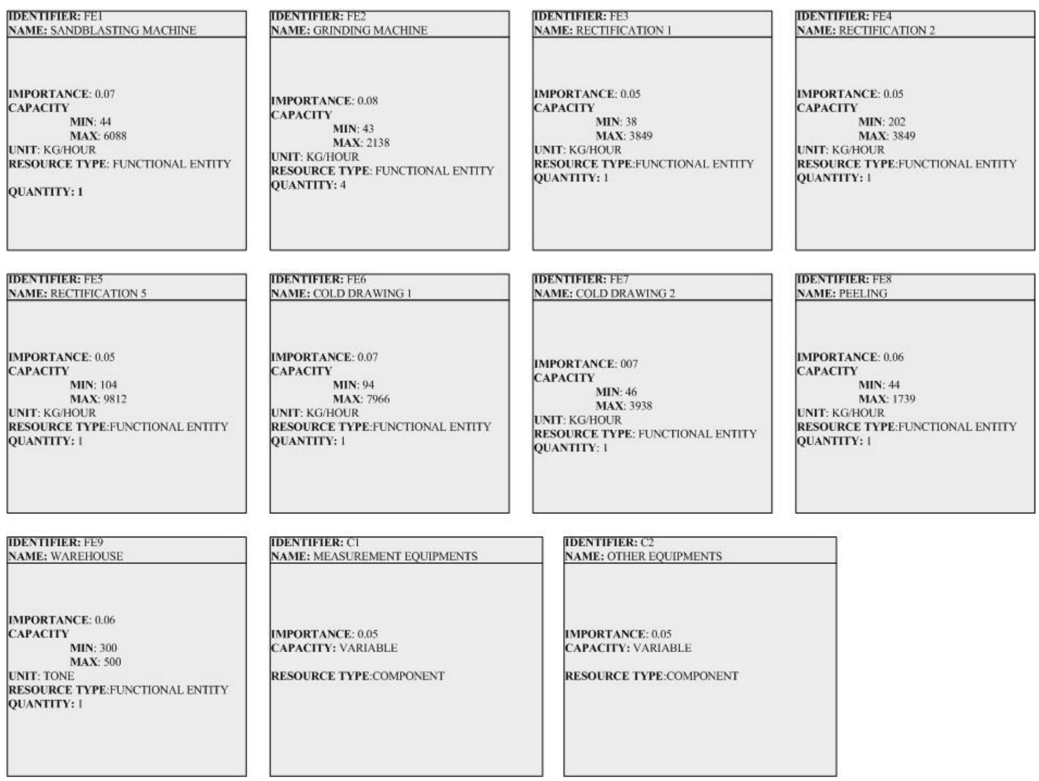


Figure 6.32 Resources defined as functional entities (FE) and components (C).

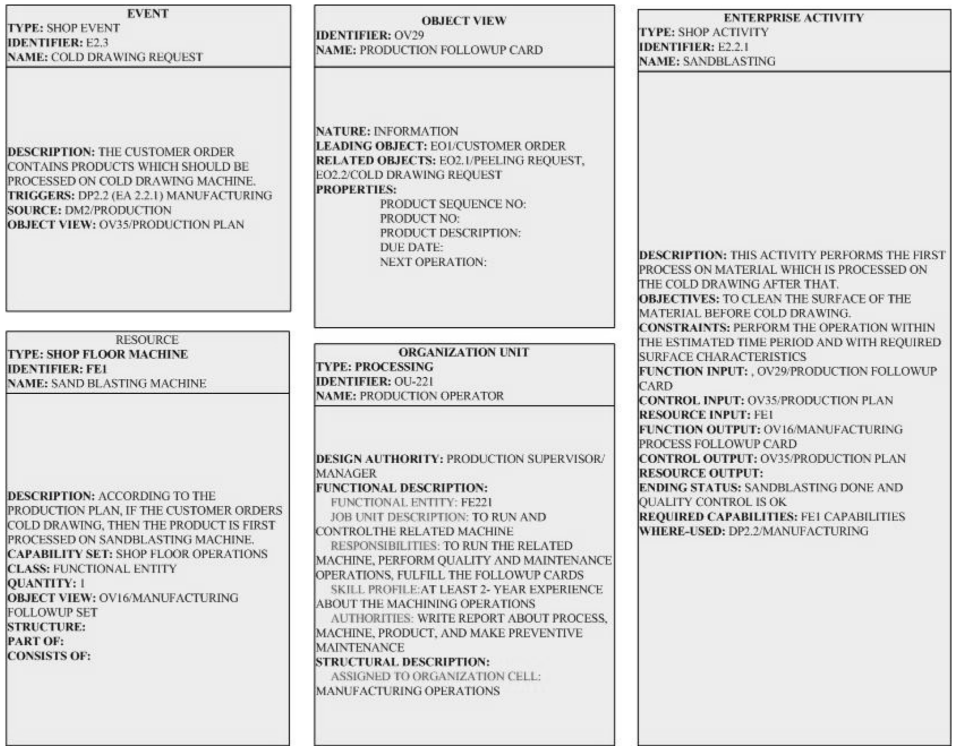


Figure 6.33 Examples of event, object view, enterprise activity, and resource definitions.

With the help of Enterprise-QFD, CIMOSA requirements modelling representations involve not only the definitions in the enterprise model, but also the importance of each entity which is analyzed within Enterprise-QFD. Enterprise-QFD collects all entities systematically in the matrices in each phase of enterprise modelling, and analyzes the relationships among these entities before transforming them into the selected reference architecture. Each entity has its own importance value concerning the relationship intensity within the enterprise, thus the entities having higher importance values become more critical. Enterprise-QFD provides a tracking opportunity from enterprise goals to the minor entity for following up. The matrices also support the enterprise engineer by representing connection points and importance weights of the relationships. Thus, the enterprise engineer can see the effects of any change within the enterprise by looking at the related phase of Enterprise-QFD.

CHAPTER SEVEN
CONCLUSIONS AND FUTURE WORK

One of the most important characteristics of today's enterprises is that they are exposed to constant change. Competitiveness and globalization force enterprises to quickly adapt to changing conditions of markets. To cope with this change, enterprises themselves need to change. This necessitates the development of a discipline organizing enterprise knowledge to manage the change expediently and professionally. Enterprises employ modelling methodologies to organize their strategic knowledge to cope with this change, which results in an enterprise model.

Enterprises should be managed as integrated systems deploying their strategies to perform their operations in order to achieve their goals. Enterprise models are the road maps to meet the predefined goals of the enterprise. Such enterprise subsystems are established for creating several models as production management for planning efficient operations, quality management for sustaining systematic management, product development systems, and financial management and cost systems. However, enterprises survive if they are managed as a whole. This necessity forces enterprise managers to consider a common model (enterprise model) providing the required infrastructure for each subsystem. In this regard, requirements discovery and analysis is the most important phase in creating an enterprise model because any mistake in the requirements discovery deteriorates the validity of the model, resulting in user dissatisfaction. Therefore, the first phase in enterprise modelling is requirements discovery. The requirements discovered are then modelled and analyzed through requirement modelling techniques. Most of these techniques in the literature are not complete enough. They are developed especially for requirements representation rather than thorough requirements analysis.

Many techniques for requirements modelling and analysis are used in software development and process design. Yet, semiformal and formal techniques are not comprehensive enough for enterprise requirements modelling. Enterprises have their own stakeholders. They create requirements, constraints and limitations for the enterprise, and thus for the enterprise model. Requirements modelling studies in the literature focus more on their representation than their discovery and analysis. However, the analysis phase is also important, and to be carried out as a process in which clear/hidden and structured/unstructured requirements are identified and analyzed.

Enterprises differ from product/ service designs in structure and scope. Enterprises are large and complicated systems performing operations in accordance with predefined goals. Enterprise goals are determined with respect to their stakeholder needs. Thus, these goals should be reflected in the enterprise architecture, and finally in the enterprise model. Consequently, requirements modelling is an important phase to be considered within the architecture.

The enterprise model should be well-matched with the enterprise goals and objectives. These goals and objectives cover not only the goals of the top management but also the goals of all entities interacting with the enterprise. Successful enterprise application and software integration need a model driven management, which is provided by the enterprise model. The integration of all systems within an enterprise application requires an enterprise model that, in turn, needs a requirements model and analysis based on the enterprise goals and objectives. Enterprise integration is generally handled from different perspectives. Therefore, a specific framework comes in view to integrate these perspectives for a complete integration process.

This study synthesized enterprise modelling, organizational characteristics and definitions, and requirements analysis and modelling concepts in a single framework. All concepts explained as the theoretical framework were examined to construct an integrated approach that proposed solutions for problems during requirements analysis phase of enterprise modelling within this framework. This thesis mainly concentrated on a requirements analysis approach, Enterprise-QFD, to manage the requirement analysis phase of enterprise modelling considering the particular framework.

Enterprise-QFD, developed based on third generation/Modern QFD by modifying existing parts and adding new matrices with respect to the enterprise modelling, is totally different from the QFD techniques developed for product/service or software design. The sequence and the content of the conventional QFD matrices were improved and redesigned for enterprise modelling. Requirement analysis based on these matrices was the major contribution of this thesis. Integration of these conversion matrices with respect to the aspects of enterprise modelling was a novel approach. The calculations were not based on traditional QFD developed in 90s, but Modern QFD which was recently being implemented by QFD experts.

Enterprise-QFD basically consists of two phases; the first phase analyzes the user's needs and gathers the enterprise goals for long term decision, and collects the necessary data for enterprise modelling components. The second phase covers the deployment process from process goals to processes, and from processes to functional requirements. Functional requirements are then deployed into informational, resource, and organizational requirements. The second phase is managed using the proposed matrices of QFD for each deployment. In each matrix, the relationships, future planning issues, technical characteristics, and related issues are analyzed and evaluated.

The major advantage of prioritizing long term goals and discovering the relationships between the long term goals and process goals is that it supports the enterprise engineer for verification of requirements. Thus, the engineer can discover a long term goal that does not have any significance compared with the others.

The evaluations and relationships defined in the QFD matrix might indicate that any process, goal, functional characteristic, or information object was related to the others more significantly. Therefore, any minor change in a design feature might affect some features more than the others. Enterprise-QFD supported the employees who are responsible for this analysis to visualize relationships to see the interactions and the affects of changes on other processes. The conflicts or correlations among characteristics or goals can also be visualized through the roof of the house. Consequently, Enterprise-QFD can support all enterprise reference architectures during the requirement definition phase by presenting an analytical approach for requirement analysis and definition even for the most complex references. The study demonstrated a novel approach through a real-case study conducted in a small business company where the proposed Enterprise-QFD was successfully applied in requirements discovery and analysis of enterprise modelling. The top management of the company and especially production and quality engineers were intensively participated in this study. All analysis results were shared with them to verify the findings and quantitative results.

This company (Guven Haddecilik San.Tic.AS) processing steel products in various shapes and dimensions. Its customers are the other companies manufacturing special-purpose machines. The company management determined the short term and long term goals. They also wished to see how the goals were supported by the processes, how they matched with each other, when any change was required in the firm, and which of the processes and characteristics could be affected by this change. Periodic visits were performed to collect data. During the current status analysis phase, all processes and the Quality System Manual were examined, a form was designed to collect primary and indirect tasks, problems, needs, and the strategic road map was provided for short and long term goals and strategies.

Enterprise modelling was a new concept for Turkey and especially for SMEs (Small and Medium Enterprises). Because of this fact, they could not see the benefits in the beginning. However, explanations about enterprise modelling and introductory meetings changed the attitude of the top management. The top management declared that only the processes defined in ISO 9000 quality management system could be handled in this project. Some of the units (personnel affairs and accounting) were discarded from the project because they were not audited by the quality system and because of their data privacy.

The activities of the project are updated after all limitations and constraints in the company were clarified. Interviews and observations were carried out with production workers and department managers by using standard forms for questions such as critical incident analysis form and gemba visit tables. All qualitative data was analyzed considering the needs of users and goals of the enterprise and then they were transformed into the enterprise modelling requirements through the proposed analysis method.

Enterprise-QFD first collects the voice of users and analyzes and classifies them by converting them into clarified need statements, then it checks if the goals are related to these needs. This phase was called preparation phase before the quantitative analysis within Enterprise-QFD. The quantitative analysis started with prioritization of long term enterprise goals. Enterprise-QFD employed Fuzzy-AHP technique for this purpose. Prioritized goals were handled to the first matrix where the long term goals were converted into the process goals, and the matrix chain was started.

In the other phases, process goals were converted into processes, and then into modelling constructs, functional requirements, informational requirements, resource requirements, and organizational requirements, respectively. Thus, desired modelling constructs were obtained through the integrated matrices analyzing the relationships, benchmarking with the competitors and planning about the future, conflictions, technical challenges and advantages of requirements, as long as the adequate data were collected. Consequently, design targets of each requirement were determined according to the final importance values. The evaluation measures were calculated as ratio scales for each evaluation using Fuzzy-AHP, and importance values were obtained as linear distribution of the evaluation values on requirements. For each requirement, a measurement unit was defined and a target was determined at the end of each matrix. At the end of the Enterprise-QFD phases, the requirement characteristics of each modelling construct was defined together with the importance values.

The relationship matrix indicated the degree of the relationship between long term goals and process goals, through this matrix, the engineer discovered a goal which was not related to any process goals, or a process goal that was not related to any long term goal in the company. The roof of the matrix might include a correlation matrix where the process goals are evaluated to observe the couples supporting each other or conflicting with each other. This part revealed the facts when any change was considered for a goal.

The gap between the structured goals of the enterprise and the general aspects of enterprise modelling were not considered in previous studies. The absence of any technique that translates the goals and characteristics of an enterprise into the enterprise modelling aspects and constructs was another motivation to propose Enterprise-QFD. With its integrated functions and stepwise application phases, Enterprise-QFD handled the requirement analysis phase as a whole project starting with the enterprise goals and ending when all inputs were ready to convert them to formal reference architectures.. However, the current techniques in the literature partially included goal setting and classification, and none of them considered a project-based tool and prioritization of requirements. Enterprise-QFD fulfilled the gaps between the requirement model and corresponding enterprise architecture.

According to the aspects of enterprise modelling, the importance values of process goals were used as the inputs for the next phase consisting of deploying the process goals to processes. After the required calculations in this matrix, weights or importance values were obtained for the processes. Thus, deploying a characteristic to another one is referred to obtaining weights of particular characteristic with respect to the evaluations, relationships, and weights of previous modelling characteristic. The major step to go into enterprise architecture aspects started after finding the weights of the processes. After the processes were determined, the calculations were continued deploying the processes in the enterprise into the functional view aspects indicating each functional object and its importance weight value. The most detailed characteristics of the enterprise were obtained in the functional aspects deployment. Since the other aspects were closely dependent on functional aspects, they were deployed on the other modelling aspects, and the objects of information view, organizational view and resource view were obtained with their importance values.

Enterprise-QFD developed the formal and informal statements and characteristics about requirements and translated them into design specifications. It was clear that the real design should be handled by the enterprise reference architectures and frameworks, i.e., CIMOSA, GERAM, GRAI, and IDEF. These architectures could provide how the formal model should be constructed, and it is about the appearance of the model. The output of the requirements analysis and modelling phase should generate the ingredients of the enterprise model, and should answer the questions: “Which processes should be in the model? What is the importance level of the processes? Which of the enterprise goals can be met by these processes? What are the interactions among the processes?” Enterprise-QFD made all these analysis and design specifications ready to be translated into the reference architectures. Furthermore, Enterprise-QFD provided the importance values of all functional, resource, organizational, and informational characteristics in the company with respect to the evaluations of customers.

Each matrix in Enterprise-QFD presented the level of relationships between columns and rows. Thus, when a change was needed in the enterprise, all matrices could be used to see how this change would affect the processes, functional, informational, resource, and organizational characteristics where the weights indicated the level of the relationships.

Enterprise Requirement Analysis process, ERA, proposed a similar framework in 2006 including goal acquisition and classification, then definition of requirements with respect to the goals. This consideration was only a framework and did not suggest any technique explaining how the goals were translated into the model. The

ERA framework in the literature could only be the starting point of requirement analysis. For a successful conversion process, a quantitative method should be integrated into ERA. The missing points in the current status of ERA could be fulfilled by a systematic analysis model. In such a model, long term and process goals should be not only classified, but also prioritized to rank projects. Furthermore, prioritized goals should be analyzed with respect to the relationships with the short term process goals. After the final goal statements were obtained, these goals should be translated into the modelling constructs, i.e., functional, informational, and organizational characteristics. Therefore some gaps were figured out between the structured goals of the enterprise and the general aspects of enterprise modelling because of the absence of any technique translating the goals and characteristics of the enterprise into enterprise modelling aspects and constructs. The proposed Enterprise-QFD approach fulfilled this gap with its integrated structure.

After the application results were collected and evaluations were completed, the findings about requirements which were ready for further modelling were translated into CIMOSA modelling constructs. Requirement statements and their relationships with each other and goals were ready to use. When all requirements corresponding to CIMOSA were ready before modelling, the transformation was very fast. The last four matrices in the analysis combined the required data according to the modelling views of CIMOSA; the only thing to do was to select the related architectural tool to write the required data within each modelling view. With the help of Enterprise-QFD, CIMOSA requirements modelling representations involved not only the definitions in the enterprise model, but also the importance of each entity which was analyzed within Enterprise-QFD. Enterprise-QFD collected all entities systematically in the matrices in each phase of enterprise modelling, and analyzed the relationships among these entities before transforming them into the selected reference architecture. Each entity had its own importance value concerning the relationship intensity within the enterprise. Thus the entities having higher importance values became more critical. Enterprise-QFD provided a tracking opportunity from enterprise goals to the minor entity for following up from the starting point to the end. The matrices also supported the enterprise engineer by representing connection points and importance weights of the relationships. Thus, the enterprise engineer could see the effects of any change within the enterprise by just looking at the related phase of Enterprise-QFD.

Enterprise-QFD is a time consuming process, if the process is handled manually. A toolbox was developed working on MS-Excel VBA modules to make the application easier. The phases of Enterprise-QFD requiring evaluations and calculations were transformed into the sheets in MS-Excel, and user forms were designed for sequential evaluations where the results are calculated automatically and recorded as structured reports in different sheets. All these partial applications were arranged in a menu added to an MS-Excel toolbar. Thus, the responsible employee could make all evaluations following the sequential steps in the menu with a basic instruction of Modern QFD.

In summary, Enterprise-QFD provided an infrastructure for sustainability of the enterprise model by providing a detailed database with importance values and relationships for all modelling components concerning the enterprise goals. It can also be integrated with other requirement modelling tools. Furthermore, some optimization techniques can be employed to achieve some target values and capacity optimization in the resource matrix.

As a conclusion, Enterprise-QFD makes all analysis and design specifications ready to be translated into some reference architectures or other requirements modelling tools, e.g. UML. The proposed Enterprise-QFD approach can be a detailed source of use cases. The outputs of Enterprise-QFD can be converted directly into an enterprise reference model, or if UML is employed, then the outputs can be translated into use case statements.

As Enterprise-QFD defines the major functionality of an enterprise in accordance with the other aspects, e.g. resource, informational, and organizational aspects, it prepares the basis for a further modelling and detailed methodologies, decision centers and applications so that all components of the enterprise work together.

Transition to an ERP system is one of the most difficult processes requiring a well-structured infrastructure for adoption to many change points within an enterprise. Enterprise-QFD contributes to the management of this change and tracking the transformation process by using the relationships discovered during the evaluations within the QFD matrices and tables which are aggregated as the rings of a chain. Enterprise requirements modelling and Enterprise-QFD enable the process integration considering the behavioural and other aspects so that ERP software can easily be adapted to a particular enterprise system.

Enterprise-QFD can be implemented in enterprises with any size. That is, the company size does not affect its framework and the components. The same tables and matrices can be used with more details. However, one should not forget that as the enterprise gets larger and more complicated, the tables and matrices get larger. This situation increases the complexity in applications. Besides, customer profiles may have higher diversity, and segmentation and analysis of customer needs may be more complicated. Resource aspects may relatively need more time if the production processes and products are complex. The detail level of each tool in Enterprise-QFD can differ with respect to the size, product or service complexity. These limitations may cause some contradictions and difficulties in managing the enterprise modelling process. At this point, efficient project management techniques may help to handle enterprise requirements analysis and the other plans and models, e.g. process models, strategic maps, policy deployment, simultaneously within the enterprise.

Enterprise-QFD has many advantages for an enterprise engineer to manage the enterprise modelling process if all integrated tools can be applied properly and in compatible with the related systems. Enterprise modelling and Enterprise-QFD should be in parallel with the strategic decision making level. Strategic plans include detailed analysis about enterprise goals, strategies, customers and stakeholders, and competitive status of the company. These predefined statements can be inputs to the requirements analysis. Processes may also be predefined in different systems, e.g. in ISO 9000 documents. Process definition is not easy to be completed properly in a short time and may be taken as a particular part of enterprise modelling. The process definitions can directly contribute the Enterprise-QFD implementation, if they have been completed before the enterprise modelling activities. The common definitions in strategic planning, enterprise modelling, and other implementations in the company should be guaranteed that they are compatible and not in contradiction. Otherwise, Enterprise-QFD cannot be beneficiary in those potential contradictions and unnecessary repetitions may result in a disadvantage for the enterprise modelling process.

Each step in Enterprise-QFD requires group decision making activities and qualitative evaluations, thus Enterprise-QFD can easily be affected by group dynamics. Dominant people may insist on their own preferences and they may keep on persuading the other during the evaluations. Top management should provide such a group platform to manage the evaluation meetings where each group member can propose his thoughts without being under pressure. Another disadvantage may arise during the qualitative evaluations throughout the linguistic variables. Customer Voice Table and Maximum Value Table statements may require general QFD training before using them because of the difficulty in retrieving the clarified items from the needs. Besides, the scale of measurement which is developed for the evaluations should be calculated with respect to the perceptions of people in the enterprise. The quantitative measures of linguistic variables should be shared with the people in the decision group and should be verified that the scale values represent their perceptions. Management and the other staff in the decision group within an enterprise should consider these conditions and potential disadvantages while implementing Enterprise-QFD.

Practitioners and academicians who recently work on enterprise modelling also consider the interoperability of all reference models. Each of the reference architectures has its own features, and sometimes the weakness of a reference may be the strength of the other one. Thus, interoperability provides both translating one reference model into another and using different parts of different reference models with each other. Therefore, requirements analysis gains another importance when interoperability is considered by this viewpoint. A detailed requirements analysis and modelling provides a well-structured base model to be translated into not only one but several reference architectures.

Enterprise QFD can also contribute to develop the enterprise ontology that expresses such an operational language and knowledge within a company. Enterprise ontology may be the continuing study after Enterprise-QFD is totally transformed to the enterprise model through the requirements modelling. Together with enterprise modelling and ontology methodologies, Enterprise-QFD may provide a systematical approach which can be suggested within the related standards about enterprise modelling and integration.

REFERENCES

183-FIPS, (1993). *Draft Federal Information Processing Standards.*

Adolph, S., Bramble, P., Cockburn, A., & Pols, A. (2002). *What Is A Quality Use Case? Patterns For Effective Use Cases.* Addison and Wesley.

Aguar, M., Coutts, I., & Weston, R. (1995). Rapid prototyping of integrated manufacturing systemsby accomplishing model-enactment. In F. V. (P.Lanet, *in Integrating Manufacturing Systems Engineering* (pp. 62-83). London: Chapman&Hall.

Akao, Y. (1988). *Quality Function Deployment: Integrating Customer Requirements into Product Design.* Portland, Oregon : Productivity Press.

Akao, Y., & Mazur, G. (2003). The leading edge in QFD: past, present and future. *International Journal of Quality & Reliability Management , 20* (1), 20-35.

Akao, Y., & Mizuno, S. (1994). *The Customer Driven Approach To Quality Planning And Deployment.* APO Press.

Al-Ahmari, A., & Ridgway, K. (1999). An integrated modelling method to support manufacturing system analysis and design. *Computers in Industry , 38*, 225-238.

AMICE. (1993). *CIMOSA: Open System Architecture for CIM, the second extended and revised version.* Berlin: Springer-Verlag .

Ang, C.-L. (1999). Enactment of IDEF0 models. *International Journal of Production Research , 37* (15), 3383-3397.

Ang, C.-L., Peng, K., & Keng Leng, G. (1999). IDEF* : a comprehensive modeling methodology for the development of manufacturing enterprise systems. *International Journal of Production Research , 37* (17), 3839-3858.

ANSI/MEA. (1994). *Committee Draft, Industrial Automation Systems, System Architecture: Framework for Enterprise Modeling.* ISO TC184 SC5 WGI.

Avison, D., & Fitzgerald, G. (2003). *Information Systems Development: Methodologies, Techniques, and Tools, Third Edition.* McGraw Hill.

Ba, S., Hinkkanen, A., & Whinston, A. (1994). Data representation and qualitative optimization-some issues in enterprise modeling. *International Conference on Industrial and Engineering Applications*, (pp. 443-446).

Bai, H., & Kwong, C. (2003). Inexact genetic algorithm approach to target values setting of engineering requiremnets in QFD. *International Journal of Production Research , 41* (16), 3861-3881.

Berio, G., & Vernadat, F. (1999). Enterprise modeling with CIMOSA: models and methods. *International Enterprise Modeling Conference* . Norway.

Berio, G., & Vernadat, F. (2001). Enterprise modelling with CIMOSA: functional and organizational aspects. *Production Planning & Control , 12* (2), 128-136.

Berio, G., & Vernadat, F. (1999). New developments in enterprise modeling. *Computers in Industry , 40*, 99-144.

Berio, G., Anaya, V., & Ortiz, A. (2004). Supporting enterprise integration through a unified enterprise modeling language.Article.

Bernus, P. (2001). Some thoughts on enterprise modeling. *Production Planning and Control , 12* (2), 110-118.

Bernus, P., & Nemes, L. (1994). A framework to define a generic enterprise reference architecture and methodology. *Proc. Third International Conference on Automation, Robotics and Computer Vision* . Singapore: (ICARCV94) .

Bernus, P., & Nemes, L. (1997). The requirements of generic enterprise reference architecture and methodology. *A. Review and Control , 21*, 125-136.

Bernus, P., Nemes, L., & Williams, T. (1995). *Architectures for Enterprise Integration; Findings of the IFAC/IFIP Task Force on Architectures for Enterprise Integration.* Chapman & Hall.

Betts, M. (1990). QFD integrated with software engineering. *2nd Symposium on Quality Function Deployment.*

Bevilacqua, M., D'Amore, A., & Polonora, F. (2004). A multi-criteria decision approach to choosing the optimal balancing-freezing system. *Journal of Food Engineering , 63*.

Boender, C., De Graan, J., & Lootsma, F. (1989). Multicriteria decision analysis with fuzzy pairwise comparisons. *Fuzzy Sets and Systems , 29*, 133-143.

Booch, C., Rumbaugh, J., & Jacobson, I. (1999). *The unified modeling language user.* Addison-Wesley.

Bossert, J. (1991). *Quality Function Deployment A Practitioner's Approach.* Milwaukee: ASQC Quality Press.

Bouyssou, D., Marchant, T., Pirlot, M., Perny, P., Tsoukias, A., & Vincke, P. (2000). *Evaluation models: A Critical Perspective.* Boston: Kluwer Inc.

BPMN. (2003). *Business Process Management Initiative. Business process modeling notation.* working draft (1.0).

Brown, S. (2004). *JAD, PD, and RAD.* Retrieved March 20, 2009, from Software Engineering Research Network, University of Calgary: http://sern.ucalgary.ca/~bowen/613/jad/

Bruno, G., & Torchiano, M. (1999). Making CIMOSA operational: the experience with the PrimeObjects Tool. *Computers in Industry , 40*, 279-291.

Buckley, J. (1985). Fuzzy hierarchical analysis. *Fuzzy Sets and Systems , 17*, 233-247.

Buuren, R. v., Jonkers, H., Iacob, M.-E., & Strating, P. (2004). *Composition Of Relations in Enterprise Architecture Models.* Enschede: Springer-Verlag Berlin Heidelberg 2004.

Büyüközkan, G., & Feyzioğlu, O. (2005). Group decision making to better respond customer needs in software development. *Computers and Industrial Engineering , 48*, 427–441.

C4ISR. (2002). Retrieved 2005, from Architecture Working Group: http://www.opengroup.org/public/arch/p4/others.htm.

Carla, R. (1999). Operational blocks for business process modeling. *Computers in Industry , 40*, 115-123.

Carmel, E., Whitaker, R., & George, J. (1993). PD and joint application design: a transatlantic comparison. *Communications of the ACM , 36* (6), 40-48.

CEN. (1994). An evaluation of CIM modelling constructs. Evaluation report of constructs for views according to ENV 40 003. *Computers in Industry , 24* (2-3), 159-236.

CEN. (1990). *ENV 40 003: Computer-integrated manufacturing - systems architecture-framework for enterprise modelling, CEN/CENELEC (also Document ISO TC184/SC5/WG1 N163).* Brussels: ISO.

Chalmeta, R., Campos, C., & Grangel, L. (2001). References architectures for enterprise integration. *Journal of Systems and Software , 57* (3), 175-191.

Chalmeta, R., Lario, F., & Ross, L. (1996). A model for reengineering project management. *The Third International Congress on Project Management.* Barcelona.

Chang, D.-Y. (1992). *Extent Analysis and Synthetic Decision, Optimization Techniques and Applications.* Singapore: World Scientific.

Chen, L., & Weng, M. (2003). Fuzzy model for exploiting Quality Function Deployment. *Mathematical and Computer Modelling , 38*, 559-570.

Chen, P. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems , 1* (1), 9-36.

Chen, Y., & Chen, L. (2005). A non-linear possibilistic regression approach to model functional relationships in product planning. *International Journal of Advanced Manufacturing Technology , 28*, 1175–1181.

Cheng, C.-H. (1996). Evaluating naval tactical missile systems by Fuzzy-AHP based on the grade value of membership function. *European Journal of Operational Research, 9*, 343-350.

Cheng, C.-H., Yang, K.-L., & Hwang, C.-L. (1999). Evaluating attack helicopters by ahp based on linguistic pairwise comparison matrices. *Computers and Operations Research , 116*, 423-435.

Chin, K.-S., Lam, J., Chan, J., Poon, K., & Yang, J. (2005). A CIMOSA presentation of an integrated product design review framework. *International Journal of Computer Integrated Manufacturing , 84* (4), 260-278.

Cho, H., & Lee, I. (1999). Integrated framework of IDEF modeling methods for structured design of shopfloor control systems. *International Journal of Computer Integrated Manufacturing , 12* (2), 113-128.

Choo, E., & Wedley, W. (2004). A common framework for deriving preference values from pairwise comparison matrices. *Computers and Operations Research , 31*, 893-908.

CIMOSA. (1999). *CIMOSA modeling framewok*. Retrieved 2005, from www.rgcp.com

CIMOSA. (1996). *Technical Baseline.* Boblingen, Germany: CIMOSA Association.

CIMOSA website. (1999). Retrieved 2005, from www.cimosa.de

Cockburn, A. (2000). *Writing Effective Use Cases.* MA: Addison-Wesley.

Crawford, A. (1994). *Advancing business concepts in a JAD workshop setting.* Englewood Cliffs, NJ: Prentice Hall.

Curtis, B., Kellner, M., & Over, J. (1992). Process modelling. *Communications of the ACM , 35* (9), 75-90.

Dahlbom, B., & Mandahl, M. (1994). A theory of information technology use. *Proceedings of the 17th IRIS* (pp. 66-77). University of Oulu.

Dardene, A., Lamsweerd, A., & Fickas, S. (1993). Goal directed requirements acqusition. *Science of Computer Programming , 20*, 3-50.

Davidson, E. (1993). An explatory study of joint application design in information systems delivery. *Proceedings of the 14th International Conference on Information Systems*, *12*, pp. 271-278.

Davies, P., & Holmes, S. (1998). Integrating rapid application development and participatory design. *IEE Proceedings-Software , 145* (4), 105-112.

Davis, A. (1993). *Software Requirements-objects, Functions and States.* PTR-Prentice-Hall.

Day, R. (1997). *Kalite Fonksiyon Yayılımı.* Marshall Inc.

DeMarco, T. (1978). *Structured Analysis and System Specification.* New York: Yourdon Inc.

Denney, R. (2005). *Succeeding with Use Cases: Working Smart To Deliver Quality.* New Jersey: Addison Wesley Professional-Pearson Education.

Dennis, A., Hayes, G., & Daniels, R. J. (1999). Business process modeling with group support systems. *Journal of Management Information Systems , 15* (4), 115-142.

Dorador, J., & Young, R. (2000). Application of IDEF0, IDEF3 and UML methodologies in the creation of information models. *International Journal of Computer Integrated Manufacturing , 13* (5), 430-445.

Doumeingst, G., Ducq, Y., & Kromm, H. (1999). Enterprise modeling techniques to improve efficiency of enterprises. *International Enterprise Modeling Conference.* Norway.

Doumeingts, G., Vallespir, B., Zanettin, M., & Chen, D. (1992). *GIM, GRAI Integrated methodology -a methodology for designing CIM systems, Version 1.0.* University of Bordeaux.

Ducq, Y., Chen, D., & Vallespir, B. (2004). Interoperability in enterprise modeling: requirements and roadmap. *Advanced Engineering Informatics* (18), 193-203.

Duenas, J. (2004). *Requirement Engineering In System Families*. Madrid, Spain: Universidad Politecnica de Madrid.

Duggana, E., & Thachenkary, C. (2004). Integrating nominal group technique and joint application development for improved systems requirements determination. *Informaiton & Management* (41), 399-411.

EA_Standards. (1999). Retrieved 2003, from EA Standards: http://www.enterprise-architecture.info/EA_Standards.htm

Eiric, P. (1992). Enterprise modeling: issues, problems and approaches. *International Conference on Enterprise Integration Modeling Technology.* South Corolina.

Elboushi, M. I., & Sherif, J. (1997). Object-oriented software design utilizing Quality Function Deployment. *Journal of Systems Software , 38*, 133-143.

Enterprise Requirements Analysis. (2006). Retrieved Feb 17, 2008, from Informatics: www.informatics.manchester.ac.uk/Comp-ISG/era/ERA_Web.html#ERA

ENV40003. (1990). Computer integrated manufacturing systems architecture framework for enterprise modelling. *CEN/CENELEC* .

Eriksson, E., & Penker, M. (2000). *Business modeling with UML.* Retrieved March 2004, from www.utm.mx/~caff/poo2.

Finkelstein, A. (1994). Requirements engineering: a review and research agenda. *Software Engineering Conference Proceedings* (pp. 10-19). Tokyo: IEEE.

Fortuna, R. (1988, June). Beyond Quality, taking SPC upstream. *Quality process* .

Fox, M., & Gruninger, M. (1998). Enterprise modeling. *AI Magazine , 19* (3), 109-121.

Francheschini, F. (2002). *Advance Quality Function Deployment.* Boca Raton: St.Lucie Press, CRC Press Company.

Frankel, D. (2003). *Model Driven Architecture : Applying MDA To Enterprise Computing*. John Wiley & Sons.

Garwin, D. (1998, July-Agust). Building learning organization. *Harvard Business Review* .

Gingele, J., Childe, S., & Miles, M. (2002). A modelling technique for re-engineering business processes controlled by ISO 9001. *Computers in Industry , 49*, 235-251.

Grady, J. (2006). *System Requirements Analysis.* Elsevier Academic Press.

Grady, J. (2007). *System Verification: proving design solution satisfies the requirements.* Elsevier Academic Press.

Greenbaum, J., & Kyng, M. (1991). *Design At Work: Cooperative Design Of Computer Systems.* Chichester, UK: Lawrence Erlbaum Associates.

Guinta, L., & Praizler, N. (1993). *The QFD book, the team approach to solving problems and satisfying customers through QFD.* New York: Amacom.

Haag, S., Raja, M., & Schkade, L. (1996). Quality Function Deployment usage in software development. *Communications of the ACM , 39* (1).

Hammer, M., & Champy, J. (1993). *Re-Engineering De Corporation.* Harper Collins.

Hauser, J., & Clausing, D. (1988, May-June). The house of quality. *Harvard Business Review* .

Hay, D. (2003). *Requirements Analysis: from Business Views to Architecture.* New Jersey: Pearson Education.

Herzwurm, G., Schokert, S., Dowie, U., & Breidung, M. (2002). Requirements engineering for mobile-commerce applications. QFD-I Germany.

Hussein, A., & Kremer, R. (2004). *What is requirements engineering? software engineering research network, University of Calgary.* Retrieved March 20, 2007, from http://sern.ucalgary.ca/courses/seng/611/W04/Alnotes611.1.

ICEIMT. (1992). *Enterprise modeling.* Retrieved 2006, from http://tools.org/EI/ICEIMT/archive/.

IDEF0. (1993, december 21). Integration definition for function modeling (IDEF0),*Computer Systems Laboratory, National Institute of Standards and Technology*.

IEEE1233. (1998). *IEEE Guide for Developing System Requirements Specifications.* IEEE-SA Standards Board.

Interop Project. (2003). Retrieved June 2005, from interoperability of enterprise Modeling Languages (UEML 2.0): www.interop.org

ISO. (1998a). *Industrial automation systems- concepts and rules for enterprise models.* Retrieved June 25, 2005, from International Standard ISO 14258, Ref No: ISO 14258:1998 (E): www.iso.org

ISO. (1990). *Reference Model for Shop Floor Production Standards.* ISO.

ISO. (1998b). Requirements for enterprise reference architectures and methodologies. ISO 15704-*ISO TC184/SC5/WG1 N423* : www.iso.org.

IWWSD. (1993). Proceedings of 7th International Workshop on Software Specification and Design. IEEE CS press.

Jackson, M. (1995). *Software Requirements and Specifications.* Addison-Wesley.

Jacobson, I. (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach.* ACM Press.

Jayaswal, B.K., Patton, P.C., Zultner, R.E. (2007). *The Design for Trustworthy Software Compilation Understanding Customer Needs: software QFD and the voice of the customer*.Prentice-Hall, URL: http://proquestcombo.safaribooksonline.com/9780132351348.

Jain, P., Mohan, P. K., & Dholakia, P. (2004, October 25). Requirements modelling (Semi-formal Specifications) for synergy distributed meeting scheduler. *CS361-Requirements Engineering-Project-II* .

Johannesson, P., & Perjons, E. (2001). Design principles for process modelling in enterprise. *Information Systems , 26*, 165-184.

Kacprzak, M., & Kaczmarczyk, A. (2006). Verification of integrated IDEF models. *Journal of Intelligent Manufacturing , 17*, 585-596.

Kahraman, C., Cebeci, U., & Ruan, D. (2004). Multi-criterion comparison of catering service companies using Fuzzy-AHP . *International Journal of Production Economics , 87*, 171-184.

Kahraman, C., Ertay, T., & Büyüközkan, G. (2006). A fuzzy optimization model for qfd planning process using analytic network Approach. *European Journal of Operational Research , 171*, 390-411.

Kapucugil Ikiz, A., & Ozdagoglu, G. (2008). Customer-driven process improvement in a shipping line company. *14th International Symposium on QFD*. Beijing: CAQ.

Kapucugil, A., Atrek, B., Ozdagoglu, A., & Tüzemen, A. (2006). Development of ideal campus settings and activities at Dokuz Eylul University. *12th International Symposium on Quality Function Deployment* (pp. 224-233). Tokyo: ICQFD.

Karsak, E. (2004). Fuzzy multiple objective programming framework to prioritize design requirements in Quality Function Deployment. *Computers & Industrial Engineering , 47* (2-3), 149-163.

Karsak, E., Sözer, S., & Alptekin, S. (2002). Product planning in Quality Function Deployment using a combined analytic network process and goal programming approach. *Computers & Industrial Engineering , 44*, 171–190.

Kim, C., Weston, R., Hodgson, A., & Lee, K. (2003). The complementary use of IDEF and UML modelling approaches. *Computers in Industry , 50*, 35-56.

Kim, K., Moskowitz, H., Dhingrra, A., & Evans, G. (2000). Fuzzy multicriteria models for Quality Function Deployment. *European Journal Of Operational Research , 121* (3), 504-518.

Kirikova, M. (2000). Explanatory capability of enterprise models. *Data & Knowledge Engineering , 33*, 119-136.

Kosanke, K., & Nell, J. (1997). Enterprise engineering and integration: building international consensus. *Proceedings of the ICEIMT'97.* Springer-Verlag.

Kosanke, K., & Nell, J. (1999). Standardisation in ISO for enterprise engineering and integration. *Computers in Industry , 40*, 311–319.

Kosanke, K., & Vernadat, F. (1998). CIMOSA—lifecycle-based enterprise. In A. Molina, A. Kusiak, & J. e. Sanches, *Handbook of Life Cycle Engineering Concepts, Models and Technologies* (pp. 181-225). Kluwer Academic Publishers.

Kosanke, K., & Zelm, M. (1999). CIMOSA modeling processes. *Computers in Industry, 40*, 141-153.

Kosanke, K., Vernadat, F., & Zelm, M. (1999). CIMOSA: Enterprise engineering and integration. *Computers in Industry , 40*, 83-97.

Kotonya, G., & Sommerville, I. (1997). *Requirements Engineering Process and Techniques* . Wiley Press.

Kotonya, G., & Sommerville, I. (1996). Requirements engineering with viewpoints. *Software Engineering Journal , 11* (1), 375-387.

Kruchten, P. (2000). *The Rational Unified Process*: *An Introduction*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Kulak, O., & Kahraman, C. (2005). Fuzzy multi-criterion selection among transportation companies using axiomatic design and analytical hierarchy process. *Information Sciences , 170*, 192-210.

Kumar, R. (2001). Managing risks in IT projects: an options perspective. *Information and management , 40* (1), 63-74.

Kuo, R., Chi, S., & Kao, S. (2002). A decision support system for selecting convenience store location through integration of Fuzzy-AHP and artifical neural networks. *Computers in Industry , 47* (2), 199-214.

Laarhoven, P., & Pedrycz, W. (1983). A Fuzzy extension of saaty's priority theory. *Fuzzy Sets and Systems , 11*, 229-241.

Ladet, P., & Vernadat, F. (1995). *Integrated Manufacturing Systems Engineering.* London: Chapman&Hall .

Lee, G., & Kusiak, A. (2001). The house of quality for design rule priority. *International Journal of Advance Manufacturing Technologies , 17*, 288-296.

Lesley, S. (2000). Quality Function Deployment and the software design environment. *12th Symposium on Quality Function Deployment.*

Leung, L., & Chao, D. (2000). On consistency and ranking of alternatives in Fuzzy-AHP . *European Journal of Operational Research , 124*, 102-113.

Levi, M., & Klapsis, M. (1999). The first step process modeler-a CIMOSA-compliant modeling. *Computers in Industry , 40*, 267-277.

Li, H., & Williams, T. (2002). Management complexity in enterprise integration projects by the PERA methodology. *Journal of Intelligent Manufacturing , 13*, 417-427.

Li, H., & Williams, T. (2000). The interconnected chain of enterprises as presented by the PERA. *Computers in Industry , 42*, 265-274.

Liles, D., & Presley, A. (1996). Enterprise modeling within an enterprise engineering framework. *Winter Simulation Conference*, (pp. -). San Diego,CA.

Lillehagen, F., & Karlsen, D. (1996). *Visiual extented enterprise engineering and operation-embedding knowledge management and work execution.* Retrieved December 20, 2005, from www.metis.no/info/white_papers/ee/index.html

Liu, S. (2005). Rating design requirements in Fuzzy Quality Function Deployment via mathematical programming approach. *International Journal Of Production Research , 43* (3), 497-513.

Lootsma, F. (1997). *Fuzzy Logic for Planning and Decision Making.* Dordrecht: Kluwer Inc.

Lopes, P. S., & Barreto, M. R. (2002). Requirements modeling with UML discussed. *Boletin Tecnico* . Escola Politecnica da USP.

Loucopoulos, P., & Kavakli, V. (1999). *Enterprise Knowledge Management*. Manchester, U.K.

Markovitz, H., & Kim, K. J. (1997). QFD Optimizer, a novice friendly QFD decision support system for optimizing product designs. *Computers and Industrial Engineering , 32* (3), 641-655.

Mayer, R., Menzel, C., Painter, M., DeWith, P., Blinn, T., & Perakath, B. (1995). *Information integration for concurrent engineering IDEF3 Process Description Capture Method Report.* Knowledge Based Systems. Inc.

Mazur, G. (2008). Current trends in Quality Function Deployment (QFD). *2nd National Symposium on QFD* (pp. 1-8). Izmir: Dokuz Eylul University.

Mazur, G. (1991). *Glenn Mazur*. Retrieved March 20, 2009, from History of QFD: www.icqfd.org

Mikhailov, L., & Singh, M. (2003). Fuzzy analytic network process and its application to the development of decision support systems. *IEEE Transactions on Systems , 33*, 33-41.

Monfared, R., & Weston, R. (1999). An Application of cimosa concepts in the development of change capable manufacturing cells. *Computers in Industry , 40*, 243-257.

Myint, S. (2003). A framework of an intelligent Quality Function Deployment (IQFD) for discrete assembly environment. *Computers & Industrial Engineering , 45*, 269–283.

Naeger, G., & Rembold, U. (1994). An integrated approach to software system planning and selection based on CIMOSA Models. *Control Engineering Practice , 3* (1), 97-103.

Nagarajan, R., Whitman, L., & Cheraghi, S. (1999). Enterprise integration. *Proceedings of The 4th Annual International Conference on Industrial Engineering Theory, Applications and Practice.* San Antonio, TX USA.

Omachonu, V., & Ross, J. (2005). *Principles of Total Quality.* Boca Raton: CRC Press.

Ortiz, A., Lario, F., & Ros, L. (1999). Enterprise integration—business processes integrated management: a proposal for a methodology to develop Enterprise Integration Programs. *Computers in Industry , 40*, 155-171.

Ortiz, A., Lario, F., Ros, L., & Hawa, M. (1999). building a production planning process with an approach based on CIMOSA and workflow management systems. *Computers in Industry , 40*, 207-219.

Ozgen, O., Kurt, G., & Ozdagoglu, G. (2006). Developing the education system: problem based learning as a new product. *12th International Symposium on Quality Function Deployment* (pp. 234-243). Tokyo: International Council for QFD.

Pantakar, K. (1995). Enterprise integration modeling: review of theory and practice. *International Journal of Computer Integrated Manufacturing , 8* (1), -.

Pardasani, A., & Chan, A. (1992). Enterprise model: a decision support tool for computer integrated manufacturing. *International Conference on Object Oriented MAnufacturing Systems.* Calgary, Canada.

Park, T., & Kim, K. (1998). Determination of an optimal set of design requirements using house of quality. *Journal Of Operations Management , 16*, 569–581.

PERA. (1999). *PERA enterprise integration website*. Retrieved 2004, URL: www.pera.org.

Petrie, C. (1992). *(eds)* Enterprise integration modeling*: Proceedings of the First International Conference* . Cambridge: MIT Press.

Plaia, A., & Carria, A. (1995). Application and assessment of IDEF3- Process flow description capture method. *International Journal of Operations Management , 15* (1), 63-73.

Pohekar, S., & Ramachandran, M. (2004). Application of multi-criteria decision aking to sustainable energy planning. *Renewable and Sustainable Energy Reviews , 8*, 365-381.

Presley, A. (1997). *A representation method to support enterprise engineering, dissertation*. Retrieved 03 04, 2007, from University of Texas at Arlington: http://arri.uta.edu/eif/diss_arp.pdf

Pressman, R. S. (1997). *Software Engineering: a practitioner's approach.* McGraw Hill.

QFDI. (2006). Academic QFD GreenBelt(R) Seminar Notes. *by Richard Zultner* . Quality Function Deployment Institute, USA.

QFDI. (2002). QFD GreenBelt(R) Seminar Notes. *by Glenn Mazur* . Quality Function Deployment Insitute, USA.

Riberio, R. (1996). Fuzzy multiple criterion decision making: a review and new preference elitication techniques. *Fuzzy Sets and Systems , 78*, 155-181.

Robertson, S., & Robertson, J. (1999). *Mastering the Requirements Process.* Harlow, UK: Addison-Wesley.

Roboam, M., Zanettin, M., & Pun, L. (1989). GRAI-IDEFO-Merise (GIM): integrated methodology to analyse and design manufacturing systems. *Computer-Integrated Manufacturing Systems , 2* (2), 82-98.

Rolland, C., Nurcan, S., & Grosz, G. (1999). Enterprise knowledge development: the process view. *Information & Management , 36*, 165-184.

Ross, D. (1977). Structured Analysis(SA): a language for communicating ideas. *IEEE Transactions on Software Engineering , 3*, 6-15.

Rossenberg, D., & Kendall, S. (1999). *Use-case Driven Object Modeling with UML: A practical approach.* Addison-Wesley.

Ruggles, R. (1995). *Why Knowledge? Why now?, Perspectives on Business Innovation, Centre.* Ernst and Young I.I.P,No. 1.

Russell, R., & Taylor, B. (2005). *Operations Management.* Prentice-Hall, New Jersey

Saaty, T. (1992). *Decision making for leaders.* Pittsburg: RWS Publications.

Saaty, T. (2001). *Decision making with dependence and feedback: Analytical Network Process.* Pittsburg: RWS Publications.

Saaty, T., & Vargas, L. (1994). *Decision making in economic, political, science and technological environments with the Analytic Hierarchy Process.* Pittsburg: RWS Publications.

Saeed, M. (2004). Web development using Quality Function Deployment. *MSc. in Information Systems supervised by Tony Lowe* .

Salvato, G., Leontaritis, I., Winstone, P., Zelm, M., Rivers-Moore, D., & Salvato, D. (1999). presentation and exchange of business models with CIMOSA-XML. *Computers in Industry , 40*, 125-139.

Sarkis, J. (1993a). Developing A strategic justification methodology for computer integrated enterprise technology using IDEF and QFD. *Proceedings of The IDEF Users Group Conference*, (pp. 154-169). College Park, Maryland.

Sarkis, J. (1998). Evaluating environmentally concious business practices. *European Journal of Operational Research , 107*, 159-174.

Sarkis, J. (1993b). Linking IDEF and QFD for use as a development tool. *Presented At The ORSA/TIMS Joint National Meeting.* Chicago.

Sarkis, J., & Liles, D. (1993). Integrating QFD and IDEF0: a case study for development of a strategic justification tool. *The Proceedings of The 1993 American Society Of Engineering Management Conference*, (pp. 128-134). Dallas, Texas.

Sarkis, J., & Talluri, S. (2004). Evaluating and selecting e-commerce software and communication systems for a supply chain. *European Journal of Operational Research , 159* (2), 318-329.

Scheer, W.-A., & Kruse, C. (1994). ARIS-framework and toolset: a comprehensive business process re-engineering methodology. *Proc. Third Int. Conf. on Automation. Robotics and Computer Vision* (pp. 327-331). Singapore: (ICARCV'94).

Scneider, G., & Winters, J. (2001). *Applying use cases-a practical guide.* Addison-Wesley.

Sheu, J.-B. (2004). A hybrid fuzzy-based approach for identifying global logistics strategies. *Transportation Research , 40*, 39-61.

Shilito, M. (1994). *Advance QFD Linking Technology to Market and Company Needs.* New York: Wiley Interscience.

Shinkawa, Y., & Matsumoto, M. (2001). Identifying the structure of business process for comprehensive enterprise modeling. *IEICE Transactions on Information and Systems. , E84-D* (2), 239-248.

Shorter, D. (1999). CEN Standardization activities related to CIMOSA. *Computers in Industry , 40*, 305-310.

Smith, A. (1996). *Enterprise data modeling*. retrieved March 18, 2005, from General Accident Insurance-Enterprise Models: http://eai.ittoolbox.com/pub/AS120301b.pdf

Snoeck, M. (1999, February). *Object-oriented Enterprise Modelling with MERODE, Leuvense.* Retrieved 2004, from Leuvense Universitaire Pers: http://www.econ.kuleuven.ac.be/tew/academic/infosys/research/merode.htm

Snoeck, M., Agarwal, R., & Basu, C. (1998). Enterprise modelling. ECOOP'98 Workshop Reader. *LNCS 1543* , 222-227.

Society, I. C. (2000). *IEEE standard 1471-2000: Recommended practice for.*

Solte, D. (1999). The OPAL Platform - a CIMOSA compliant execution environment. *Computers in Industry , 40*, 293-303.

Sommerville, I., & Sawyer, P. (1997). Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering , 3*, 101-30.

Stamatis, D. (2003). *Six Sigma and Beyond.* Boca Raton: St. Lucie Press, CRC Press Company.

Stapleton, J. (1997). *DSDM - dynamic systems development method: the method in practice.* Harlow-England: Addison-Wesley.

Stolfa, S., & Vondrak, I. (2004). An explanation of automized transformation procedure from business processes to use case diagrams. *Proceedings of ISIM*, (pp. 101-107). Radhost .

Swenson, K., & Irwin, K. (1995). Workflow technology: tradeoffs for business processes. *Conference on Organisational Computing Systems COOCS 95.* California.

Tang, J., Fung, R., Xu, B., & Wang, D. (2002). A new approach to Quality Function Deployment planning with financial consideration. *Computers & Operations Research , 29*, 1447-1463.

Taylor, B. (2004). *Introduction to Management Science.* New Jersey: Pearson Education.

TEAF. (2000). *Department of Treasury, Chief Information Officer Council*. Retrieved 2005, from Treasury Enterprise Architecture Framework, version1.

Temponi, C., Yen, J., & Tiao, W. (1999). House of quality: a fuzzy logic-based requirements analysis. *European Journal of Operational Research , 117*, 340-354.

Terninko, J. (1997). *Step-By-Step QFD, Customer Driven Product Design.* St.Lucia Press.

Tesfamariam, D., & Lindberg, B. (2005). Aggregate analysis of manufacturing systems using system dynamics and ANP. *Computers and Industrial Engineering , 49*, 98-117.

Tham, K. (2000). *CIMOSA enterprise modeling*. Retrieved 05 2004, from Enterprise Integration Laboratory, Toronto University: http://eil.utoronto.ca/entmethod/cimosa/cim.html

Tham, K. (2001). *PERA: enterprise modeling*. Enterprise Integration Laboratory, University of Toronto.

Tolga, E., Demircan, M., & Kahraman, C. (2005). Operating System selection using fuzzy replacement analysis and Analytic Hierarchy Process. *International Journal of Production Economics , 97*, 89-117.

UEML. (2001). Retrieved June 2005, from unified enterprise modeling language, Thematic Network - Contract no: IST – 2001 – 34229: www.ueml.org

Vairaktarakis, G. (1999). Optimization tools for design and marketing of new/improved products using the house of quality. *Journal of Operations Management , 17* (6), 645–663.

Venugopalan, T. (2003). Development of a framework for enterprise modeling. *Thesis Submitted to the Department of Industrial Engineering, Mississippi State University,* . Mississippi, US.

Vernadat, F. (1996). *Enterprise Modeling and Integration: Principles and Applications.* London: Chapman&Hall.

Vernadat, F., & Zelm, M. (1993). Advance modeling approach to cim systems. in f. m.cotsatis, *Advances in Factories of the Future, CIM and Robotics* (pp. 77-89). Amsterdam: Elsevier.

Vliet, H.-V. (2007). *Software Engineering: Principles and Practice.* Wiley.

Wahlander, C., Nilsson, M., & Skoog, A. (1998). Introduction to business model language & smartsync model manager. Copyright Viewlocity.

Ward, P., & Mellor, S. (1985). *Structured Development for Real Real-Time Systems.* New Jersey: Prentice-Hall.

Weston, R. (1999). Reconfigurable, component-based systems and the role of enterprise engineering concepts. *Computers in Industry , 40*, 321-343.

Williams, T. (1994). The purdue enterprise reference architecture. *Computers in Industry , 24* (2-3), 141-158.

Williams, T. (1998). The purdue enterprise reference architecture and methodology. In *Handbook of Life Cycle Engineering: Concepts, Tools, and Techniques.*

Wilson, M., Aguiar, C., & Edwards, J. (1999). Achieving Manufacturing business integration through the combined formalisms of CIMOSA and Petri Nets. *International Journal of Production Research , 37* (8), 1767-1786.

Withman, L., & Huff, B. (2001). On the use of enterprise models. *The International Journal of Flexible Manufacturing Systems , 13*, 195-208.

Wortmann, J. C., Hegge, H. M., & Rolefes, S. (2000). Embedding enterprise software in extended enterprise models. *Computers in Industry , 42*, 231-243.

FAA (Federal Aviation Administration) (1995). U.S. Depertmant of Transportation. Retrieved February 20, 2005, from Office of Information Technology Integrated Product Team for Information Technology Services, Business Process Improvement (reengineering) version 1.0: www.faa.gov./ait/handbook/chap4.htm

Yang, K., & El-Haik, B. (2009). *Design for Six Sigma.* New York: McGraw-Hill.

Yang, Y., Wang, S., Dulaimi, M., & Low, P. (2003). A fuzzy Quality Function Deployment system for buildable design. *Automation In Construction , 12*, 381– 393.

Yenginol, F. (2002). Neden kalite fonksiyon 'göçerimi'? *I.Ulusal Kalite Fonksiyon Gocerimi Sempoyumu Bildiriler Kitabi* (pp. 50-62). Izmir: Dokuz Eylul University.

Yoshikawa, H., & Goossenaerts, J. (1993). Information infrastructure systems for manufacturing. *Proceedings of the JSPE/IFIP tc/WG5.3 Workshop on the Design of Information Infrastructure Systems for Manufacturing.* Tokyo: North-Holland.

Yu, C.-S. (2002). A GP-AHP method for solving group decision-making Fuzzy-AHP problems. *Computers and Operations Research , 29*, 1969-2001.

Yu, J.-R., & Cheng, S.-J. (2007). An integrated approach for deriving priorities in Analytic Network Process. *European Journal of Operational Research , 180*, 1427-1432.

Zachman, J. (1987). A framework for information systems architecture. *IBM Systems Journal , 26* (3), 276-292.

Zakarian, A., & Kosiak, A. (2001). Process analysis and reengineering. *Computers and Industrial Engineering , 41*, 135-150.

Zave, P. (1994). Call for papers and associated classification scheme: *IEEE International Symposium on Requirements Engineering*-1995.

Zave, P., & Jackson, M. (1995). Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology , 6* (1), 1-30.

Zelm, M., Vernadat, F., & Kosanke, K. (1995). CIMOSA business modeling process. *Computers in Industry , 27*, 123-142.

Zhou, M. (1998). Fuzzy logic and optimization models for implementing QFD. *Computers and Industrial Engineering , 35* (1-2), 237-240.

Zhu, K.-J., Jing, Y., & Chang, D.-Y. (1999). A discussion on extent analysis method and applications of Fuzzy-AHP . *European Journal of Operational Research , 116*, 450-456.

Zrymiak, D. (2003). Software quality function deployment, modifying the "house of quality" for software. *ASQ-CQMGR, CSQE, CQE, CQA*.

Zultner, R. (1995). Business process reengineering with Quality Function Deployment, Process innovation for software development. *7th Symposium on Quality Function Deployment.*

Zultner, R. E. (1997). Project QFD, managing software development projects better with Blitz QFD. *9th Symposium on Quality Function Deployment.*

Zultner, R. (2000). Software QFD, a silver bullet for software development. *12th Symposium on Quality Function Deployment.*

Zultner, R. (1990). Software Quality Deployment, adopting QFD to software. *2nd Symposium on QFD.*

APPENDICES

APPENDIX 1: ENTERPRISE- QFD TABLES

Critical Incidence Question Form

Sayın Yetkili,

Dokuz Eylül Üniversitesi Endüstri Mühendisliği Bölümü'nde Doktora yapmaktayım. Doktora Tez konusu olarak **"Kurum Modellemesi ve Kurum Mühendisliği"** konusunda araştırma yapmaktayım. Kurum modeli, özetle, mevcut durumu ve süreçleri daha iyi yönetebilmek amacıyla kurumu tüm ayrıntılarıyla bir mimari yapı gibi anlatan, ayrıntılı sistem görünümüdür. Araştırmalarım doğrultusunda bir proje geliştirdim ve bu projeyi Güven Haddecilik Tic. A.Ş. 'de araştırma olanağı buldum. Projenin etkin olarak geliştirilmesi ve uygulanması ancak sağlıklı ve eksiksiz veri elde etmekle mümkündür. Kurumun çalışanları olarak en sağlıklı bilgi sizin şimdiye kadar yaşadığınız tecrübelerdir. Bu form, çalıştığınız kurum içindeki yerinizi, genel olarak sorumlu olduğunuz işleri ve yaşadığınız kritik olayları öğrenmek için hazırlanmıştır. Formda yazan bilgilerin gizliliği esastır. Zaman ayırdığınız için şimdiden teşekkür eder mutlu günler dilerim.

Formu doldurduktan sonra elden çıktı ya da dijital olarak teslim edebilir, guzin.kavrukkoca@deu.edu.tr adresine ekli dosya olarak gönderebilirsiniz.

ADINIZ:

KURUMDAKİ GÖREV ÜNVANINIZ:

1) **En sık yaptığınız işler:**
2) **Sürekli olmayan, ancak ara sıra sorumlu olduğunuz/ilgilendiğiniz işler:**
3) **Çok yoğun bir gününüzü düşünün, sabahtan çıkış saatinize kadar bu gününüzü anlatır mısınız? (Okulda eskiden "tatilinizi nasıl geçirdiniz" konulu bir kompozisyonu yazdığınız gibi)**
4) **Kurumda yaşadığınız olumsuz bir olayı ya da olayları ve bu olaylardan sizin işinizin nasıl etkilendiğini anlatır mısınız?**
5) **Kurumda yaşadığınız olumlu bir olayı ya da olayları ve bu olaylardan sizin işinizin nasıl etkilendiğini anlatır mısınız?**
6) **Sorumluluğunuzdaki işleri yerine getirmenizi engelleyen kısıtlar, problemler var mı? Varsa bu problemleri ve bunlar için çözüm önerilerinizi listeleyiniz.**
7) **Çalıştığınız kurum ve işleriniz hakkında yukarıdakiler dışında belirtmek istedikleriniz:**

DESTEĞİNİZ İÇİN TEŞEKKÜRLER

Example for GEMBA Visit Table:

GEMBA ZİYARET TABLOSU					
Görüşülen kişi: Kalite Sistem Müdürü **Tarih:** 12.09.2006 **Email:** aylin.akal@guvencelik.com **Yer:** Ofis					
Karakteristikler: Yüksek tempolu çalışma koşulları, çoklu iş sorumlulukları, ofis ortamında ve üretim ortamında yoğun gözlem ve değerlendirmeler					
Ortam Karakteristikleri : Rahat ve genel anlamda hoş bir atmosferi olan ofis ortamı.					
Süreç	**Gözlemler**	**İfadeler**	**Dokumanlar**	**Notlar**	**Net İfade**
Kalite kontrol plan ve raporları	Her gelen malzeme için analiz sertifikaları hazırlanıyor. Düzenli üretim planı hazırlanmıyor, çoğunlukla gerekmiyor. Yeniden işlemeler olabiliyor, ancak malzeme atılması söz konusu değil Hataların önlenmesi için kontroller sıklaştırılıyor, ara kontrollere önem veriliyor.	Giriş kalite kontrolleri, final kalite kontrolleri, yurt dışından gelen malzemelerin tespiti ve analizi.	Malzeme sertifikaları ve analiz raporları	Üretim personeli ile iletişim ve hiyerarşi problemi yaşanabiliyor	Malzeme sertifikaları Malzeme Analizleri Süreç içi kontroller Üretim takipleri Ayrıntılı Bakım planı
Kalite Güvence, Kalite Yönetim sistemi	Üretim personeli için haftalık eğitim toplantıları, üst yönetim ile değerlendirme toplantıları.	Süreç takipleri, üretim takip raporları, sistem ile ilgili revizyonların yapılması, periyodik iç denetimler. makinaların yaşı oldukça büyük, çoğunlukla tam kapasite çalışıyor. Kısa dönemde yatırım planı yok.	Kalite Sistem el kitabı, prosedürler, süreç tanımları	Mevcut sistemin işleyişinde genelde problem yok, ancak gelişmeye açık noktaları var, üretim personelinin sisteme adaptasyonu konusunda problemler yaşanabiliyor, işçi turnoverı yüksek olabiliyor.	Geliştirilmiş kalite yönetim sistemi Haftalık Personel Eğitim toplantıları Haftalık Yönetim değerlendirme Kalifiye işçi istihdamı Düşük işçi sirkülasyonu

GEMBA ZİYARET TABLOSU (DEVAMI)					
Görüşülen kişi: Kalite Sistem Müdürü **Tarih:** 12.09.2006 **Email:** aylin.akal@guvencelik.com **Yer:** Ofis					
Karakteristikler: Yüksek tempolu çalışma koşulları, çoklu iş sorumlulukları, ofis ortamında ve üretim ortamında yoğun gözlem ve değerlendirmeler					
Ortam Karakteristikleri : Rahat ve genel anlamda hoş bir atmosferi olan ofis ortamı.					
Süreç	**Gözlemler**	**İfadeler**	**Dokumanlar**	**Notlar**	**Net İfade**
Bazı satın almalar	Satın alınan malzemelerde çatlak, ölçü hatası, boy hatası gibi problemler yaşanabiliyor	Satın alma süreçlerinden bazıları diğer süreç sorumluları ile paylaşılıyor			Daha izole edilmiş organizasyon yapısı
Eğitim planı hazırlama		Eğitim ihtiyaçları ve talepler doğrultusunda eğitim planı hazırlanıyor.			Planlı eğitimler
Yurt dışı ile ilişkiler		Yurt dışındaki ilişkili firmalarla ithalat-ihracat konusunda yazışmalar ve gümrük işleri			İthalat-İhracat konusunda ilgili ve ara firmalar ile iletişim
		Kalite sistem yönetici olarak yıllık hedefler ve hedeflere ulaşmak için izlenecek eylem planı hazırlanıyor			Tanımlı birim hedefleri ve eylem planı

APPENDIX 2: EXCEL MODULES FOR ENTERPRISE-QFD

MS-Excel Measurement Calculations Through Fuzzy-AHP (e.g. Relationship Matrix)

```
Private Sub CommandButton1_Click()
Dim i As Byte
Dim j As Byte
Sheets("DRAFT").Activate
Cells.Clear
Range("c3:az10").Select
Selection.NumberFormat = "0.000"
For i = 1 To 15
    For j = 1 To 15
    Cells(i, j) = ""
    Next j
Next i
Cells(3, 4) = TextBox1.Value
Cells(3, 5) = TextBox2.Value
Cells(3, 6) = TextBox3.Value
Cells(4, 5) = TextBox11.Value
Cells(4, 6) = TextBox12.Value
Cells(5, 6) = TextBox20.Value
'satır ve sütunlara ölçüt numaralarının yazdırılması
i = 2
Cells(i, 3) = "ÖLÇÜT 1"
Cells(i, 4) = "ÖLÇÜT 2"
Cells(i, 5) = "ÖLÇÜT 3"
Cells(i, 6) = "ÖLÇÜT 4"
Cells(3, i) = "ÖLÇÜT 1"
Cells(4, i) = "ÖLÇÜT 2"
Cells(5, i) = "ÖLÇÜT 3"
```

```
Cells(6, i) = "ÖLÇÜT 4"
'bilgi almak isteyenler için üçlü bulanık sayıların tanıtılması
Cells(11, 2) = "GİRİLECEK DEĞER"
Cells(11, 3) = "l DEĞERİ"
Cells(11, 4) = "m DEĞERİ"
Cells(11, 5) = "u DEĞERİ"
Cells(11, 6) = "DEĞERİN SÖZEL KARŞILIĞI"
Cells(12, 2) = 4
Cells(12, 3) = 3.5
Cells(12, 4) = 4
Cells(12, 5) = 4.5
Cells(12, 6) = "KESİNLİKLE DAHA ÖNEMLİ"
Cells(13, 2) = 3
Cells(13, 3) = 2.5
Cells(13, 4) = 3
Cells(13, 5) = 3.5
Cells(13, 6) = "DAHA ÖNEMLİ"
Cells(14, 2) = 2
Cells(14, 3) = 1.5
Cells(14, 4) = 2
Cells(14, 5) = 2.5
Cells(14, 6) = "ÖNEMLİ"
Cells(15, 2) = 2 / 3
Cells(15, 3) = 2 / 3
Cells(15, 4) = 1
Cells(15, 5) = 1.5
Cells(15, 6) = "AZ ÖNEMLİ"
Cells(16, 2) = 1
Cells(16, 3) = 1
Cells(16, 4) = 1
Cells(16, 5) = 1
Cells(16, 6) = "EŞİT ÖNEME SAHİP"
```

```
Cells(17, 2) = 0.5
Cells(17, 3) = 0.4
Cells(17, 4) = 0.5
Cells(17, 5) = 2 / 3
Cells(17, 6) = "ÖNEMLİ"
Cells(18, 2) = 1 / 3
Cells(18, 3) = 2 / 7
Cells(18, 4) = 1 / 3
Cells(18, 5) = 0.4
Cells(18, 6) = "DAHA ÖNEMLİ"
Cells(19, 2) = 0.25
Cells(19, 3) = 2 / 9
Cells(19, 4) = 0.25
Cells(19, 5) = 2 / 7
Cells(19, 6) = "KESİNLİKLE DAHA ÖNEMLİ"

'ölçüt sayısını kullanıcının belirtmesine gerek kalmadan _
tespit edip ilgili komut satırına gitmek için

'1. ölçüt ile 2. ölçütün kesişimindeki değere ilişkin üçlü _
bulanık sayıların hesap için (3,23); (3,24); (3,25) hücrelerine _
atanması
For i = 2 To 4
    For j = 3 To 5
        If Cells(3, 4) = i Then
        Cells(3, j + 20) = Cells(16 - i, j)
        ElseIf Cells(3, 4) = 0.667 Then
        Cells(3, j + 20) = Cells(15, j)
        ElseIf Cells(3, 4) = 1 Then
        Cells(3, j + 20) = Cells(16, j)
        ElseIf Cells(3, 4) = Round(1 / i, 3) Then
        Cells(3, j + 20) = Cells(i + 15, j)
```

```
        End If
    Next j
Next i

'1. ölçüt ile 3. ölçütün kesişimindeki değere ilişkin üçlü _
bulanık sayıların hesap için (3,26); (3,27); (3,28) hücrelerine _
atanması
For i = 2 To 4
    For j = 3 To 5
        If Cells(3, 5) = i Then
        Cells(3, j + 23) = Cells(16 - i, j)
        ElseIf Cells(3, 5) = 0.667 Then
        Cells(3, j + 23) = Cells(15, j)
        ElseIf Cells(3, 5) = 1 Then
        Cells(3, j + 23) = Cells(16, j)
        ElseIf Cells(3, 5) = Round(1 / i, 3) Then
        Cells(3, j + 23) = Cells(i + 15, j)
        End If
    Next j
Next i
'1. ölçüt ile 4. ölçütün kesişimindeki değere ilişkin üçlü _
bulanık sayıların hesap için (3,29); (3,30); (3,31) hücrelerine _
atanması
For i = 2 To 4
    For j = 3 To 5
        If Cells(3, 6) = i Then
        Cells(3, j + 26) = Cells(16 - i, j)
        ElseIf Cells(3, 6) = 0.667 Then
        Cells(3, j + 26) = Cells(15, j)
        ElseIf Cells(3, 6) = 1 Then
        Cells(3, j + 26) = Cells(16, j)
        ElseIf Cells(3, 6) = Round(1 / i, 3) Then
```

```
        Cells(3, j + 26) = Cells(i + 15, j)
        End If
    Next j
Next i

'2. ölçüt ile 3. ölçütün kesişimindeki değere ilişkin üçlü _
bulanık sayıların hesap için (4,26); (4,27); (4,28) hücrelerine _
atanması

For i = 2 To 4
    For j = 3 To 5
        If Cells(4, 5) = i Then
        Cells(4, j + 23) = Cells(16 - i, j)
        ElseIf Cells(4, 5) = 0.667 Then
        Cells(4, j + 23) = Cells(15, j)
        ElseIf Cells(4, 5) = 1 Then
        Cells(4, j + 23) = Cells(16, j)
        ElseIf Cells(4, 5) = Round(1 / i, 3) Then
        Cells(4, j + 23) = Cells(i + 15, j)
        End If
    Next j
Next i

'2. ölçüt ile 4. ölçütün kesişimindeki değere ilişkin üçlü _
bulanık sayıların hesap için (4,29); (4,30); (4,31) hücrelerine _
atanması

For i = 2 To 4
    For j = 3 To 5
        If Cells(4, 6) = i Then
        Cells(4, j + 26) = Cells(16 - i, j)
        ElseIf Cells(4, 6) = 0.667 Then
```

```
        Cells(4, j + 26) = Cells(15, j)
        ElseIf Cells(4, 6) = 1 Then
        Cells(4, j + 26) = Cells(16, j)
        ElseIf Cells(4, 6) = Round(1 / i, 3) Then
        Cells(4, j + 26) = Cells(i + 15, j)
        End If
    Next j
Next i

'3. ölçüt ile 4. ölçütün kesişimindeki değere ilişkin üçlü _
bulanık sayıların hesap için (5,29); (5,30); (5,31) hücrelerine _
atanması

For i = 2 To 4
    For j = 3 To 5
        If Cells(5, 6) = i Then
        Cells(5, j + 26) = Cells(16 - i, j)
        ElseIf Cells(5, 6) = 0.667 Then
        Cells(5, j + 26) = Cells(15, j)
        ElseIf Cells(5, 6) = 1 Then
        Cells(5, j + 26) = Cells(16, j)
        ElseIf Cells(5, 6) = Round(1 / i, 3) Then
        Cells(5, j + 26) = Cells(i + 15, j)
        End If
    Next j
Next i

'(3,20);(3,21);(3,22);(4,23);(4,24);(4,25);(5,26);(5,27);(5,28) _
;(6,29);(6,30);(6,31) hücrelerine aynı _
ölçütler olduğu için 1 değerlerinin atanması

'alt üçgendeki değerlerin otomatik olarak hesaplanması
```

```
Cells(4, 20) = 1 / Cells(3, 25)
Cells(4, 21) = 1 / Cells(3, 24)
Cells(4, 22) = 1 / Cells(3, 23)

For j = 20 To 22
Cells(3, j) = 1
Next j
For j = 23 To 25
Cells(4, j) = 1
Next j

For j = 26 To 28
Cells(5, j) = 1
Next j
    Cells(5, 20) = 1 / Cells(3, 28)
    Cells(5, 21) = 1 / Cells(3, 27)
    Cells(5, 22) = 1 / Cells(3, 26)
    Cells(5, 23) = 1 / Cells(4, 28)
    Cells(5, 24) = 1 / Cells(4, 27)
    Cells(5, 25) = 1 / Cells(4, 26)

    For j = 29 To 31
    Cells(6, j) = 1
    Next j
    Cells(6, 20) = 1 / Cells(3, 31)
    Cells(6, 21) = 1 / Cells(3, 30)
    Cells(6, 22) = 1 / Cells(3, 29)
    Cells(6, 23) = 1 / Cells(4, 31)
    Cells(6, 24) = 1 / Cells(4, 30)
    Cells(6, 25) = 1 / Cells(4, 29)
    Cells(6, 26) = 1 / Cells(5, 31)
```

```
    Cells(6, 27) = 1 / Cells(5, 30)
    Cells(6, 28) = 1 / Cells(5, 29)

'her satıra ilişkin l, m, u değerlerinin hesaplanması

l1 = Cells(3, 20) + Cells(3, 23) + Cells(3, 26) + Cells(3, 29)
m1 = Cells(3, 21) + Cells(3, 24) + Cells(3, 27) + Cells(3, 30)
u1 = Cells(3, 22) + Cells(3, 25) + Cells(3, 28) + Cells(3, 31)
l2 = Cells(4, 20) + Cells(4, 23) + Cells(4, 26) + Cells(4, 29)
m2 = Cells(4, 21) + Cells(4, 24) + Cells(4, 27) + Cells(4, 30)
u2 = Cells(4, 22) + Cells(4, 25) + Cells(4, 28) + Cells(4, 31)
l3 = Cells(5, 20) + Cells(5, 23) + Cells(5, 26) + Cells(5, 29)
m3 = Cells(5, 21) + Cells(5, 24) + Cells(5, 27) + Cells(5, 30)
u3 = Cells(5, 22) + Cells(5, 25) + Cells(5, 28) + Cells(5, 31)
l4 = Cells(6, 20) + Cells(6, 23) + Cells(6, 26) + Cells(6, 29)
m4 = Cells(6, 21) + Cells(6, 24) + Cells(6, 27) + Cells(6, 30)
u4 = Cells(6, 22) + Cells(6, 25) + Cells(6, 28) + Cells(6, 31)

'genel toplamların alınması

l = u1 + u2 + u3 + u4
m = m1 + m2 + m3 + m4
u = l1 + l2 + l3 + l4

lx1 = l1 / l
mx1 = m1 / m
ux1 = u1 / u
lx2 = l2 / l
mx2 = m2 / m
ux2 = u2 / u
lx3 = l3 / l
mx3 = m3 / m
```

```
ux3 = u3 / u
lx4 = l4 / l
mx4 = m4 / m
ux4 = u4 / u
'1. ölçütün değerleri
If mx1 >= mx2 Then
Cells(3, 50) = 1
ElseIf lx2 >= ux1 Then
Cells(3, 50) = 0
Else
Cells(3, 50) = (lx2 - ux1) / ((mx1 - ux1) - (mx2 - lx2))
End If

If mx1 >= mx3 And mx3 <> 0 Then
Cells(3, 51) = 1
ElseIf lx3 >= ux1 And mx3 <> 0 Then
Cells(3, 51) = 0
ElseIf mx3 <> 0 Then
Cells(3, 51) = (lx3 - ux1) / ((mx1 - ux1) - (mx3 - lx3))
Else
End If

If mx1 >= mx4 And mx4 <> 0 Then
Cells(3, 52) = 1
ElseIf lx4 >= ux1 And mx4 <> 0 Then
Cells(3, 52) = 0
ElseIf mx4 <> 0 Then
Cells(3, 52) = (lx4 - ux1) / ((mx1 - ux1) - (mx4 - lx4))
Else
End If

'2. ölçütün değerleri
```

```
If mx2 >= mx1 Then
Cells(4, 50) = 1
ElseIf lx1 >= ux2 Then
Cells(4, 50) = 0
Else
Cells(4, 50) = (lx1 - ux2) / ((mx2 - ux2) - (mx1 - lx1))
End If

If mx2 >= mx3 And mx3 <> 0 Then
Cells(4, 51) = 1
ElseIf lx3 >= ux2 And mx3 <> 0 Then
Cells(4, 51) = 0
ElseIf mx3 <> 0 Then
Cells(4, 51) = (lx3 - ux2) / ((mx2 - ux2) - (mx3 - lx3))
Else
End If

If mx2 >= mx4 And mx4 <> 0 Then
Cells(4, 52) = 1
ElseIf lx4 >= ux2 And mx4 <> 0 Then
Cells(4, 52) = 0
ElseIf mx4 <> 0 Then
Cells(4, 52) = (lx4 - ux2) / ((mx2 - ux2) - (mx4 - lx4))
Else
End If

'3. ölçütün değerleri

If mx3 >= mx1 And mx3 <> 0 Then
Cells(5, 50) = 1
ElseIf lx1 >= ux3 And mx3 <> 0 Then
```

```
Cells(5, 50) = 0
ElseIf mx3 <> 0 Then
Cells(5, 50) = (lx1 - ux3) / ((mx3 - ux3) - (mx1 - lx1))
Else
End If

If mx3 >= mx2 And mx3 <> 0 Then
Cells(5, 51) = 1
ElseIf lx2 >= ux3 And mx3 <> 0 Then
Cells(5, 51) = 0
ElseIf mx3 <> 0 Then
Cells(5, 51) = (lx2 - ux3) / ((mx3 - ux3) - (mx2 - lx2))
Else
End If

If mx3 >= mx4 And mx3 <> 0 And mx4 <> 0 Then
Cells(5, 52) = 1
ElseIf lx4 >= ux3 And mx3 <> 0 And mx4 <> 0 Then
Cells(5, 52) = 0
ElseIf mx3 <> 0 And mx4 <> 0 Then
Cells(5, 52) = (lx4 - ux3) / ((mx3 - ux3) - (mx4 - lx4))
Else
End If

'4. ölçütün değerleri

If mx4 >= mx1 And mx4 <> 0 Then
Cells(6, 50) = 1
ElseIf lx1 >= ux4 And mx4 <> 0 Then
Cells(6, 50) = 0
ElseIf mx4 <> 0 Then
Cells(6, 50) = (lx1 - ux4) / ((mx4 - ux4) - (mx1 - lx1))
```

```
End If

If mx4 >= mx2 And mx4 <> 0 Then
Cells(6, 51) = 1
ElseIf lx2 >= ux4 And mx4 <> 0 Then
Cells(6, 51) = 0
ElseIf mx4 <> 0 Then
Cells(6, 51) = (lx2 - ux4) / ((mx4 - ux4) - (mx2 - lx2))
Else
End If

If mx4 >= mx3 And mx3 <> 0 And mx4 <> 0 Then
Cells(6, 52) = 1
ElseIf lx3 >= ux4 And mx3 <> 0 And mx4 <> 0 Then
Cells(6, 52) = 0
ElseIf mx3 <> 0 And mx4 <> 0 Then
Cells(6, 52) = (lx3 - ux4) / ((mx4 - ux4) - (mx3 - lx3))
Else
End If

'en küçük olanın seçilmesi

For i = 3 To 10
Cells(i, 60) = Cells(i, 50)
Next i

For i = 3 To 10
   For j = 50 To 56
      If Cells(i, j) <= Cells(i, 60) And Cells(i, j) <> "" Then
      Cells(i, 60) = Cells(i, j)
      Else: Cells(i, 60) = Cells(i, 60)
      End If
```

```
    Next j
Next i

'normalize etmek için bu değerlerin toplamının alınması

sutuntop = 0
For i = 3 To 10
sutuntop = Cells(i, 60) + sutuntop
Next i

'normalize edilmiş ağırlıklar

For i = 3 To 10
Cells(i, 11) = Cells(i, 60) / sutuntop
Next i

'Sheet2.Range("b1:b4") = Sheet1.Range("k3:k6")
For i = 2 To 5
Sheet2.Cells(i, 2) = Sheet10.Cells(i + 1, 11)
Next i

Sheet2.Range("a1") = "RELATIONSHIP"
Sheet2.Range("a2") = "VERY STRONG RELATIONSHIP"
Sheet2.Range("a3") = "STRONG RELATIONSHIP"
Sheet2.Range("a4") = "WEAK RELATIONSHIP"
Sheet2.Range("a5") = "NO RELATIONSHIP"

Sheet2.Columns.EntireColumn.AutoFit

If Cells(3, 5) = "" Then
    For i = 5 To 10
    Cells(i, 11) = ""
```

```
    Next i
ElseIf Cells(3, 6) = "" Then
    For i = 6 To 10
    Cells(i, 11) = ""
    Next i
ElseIf Cells(3, 7) = "" Then
    For i = 7 To 10
    Cells(i, 11) = ""
    Next i
ElseIf Cells(3, 8) = "" Then
    For i = 8 To 10
    Cells(i, 11) = ""
    Next i
ElseIf Cells(3, 9) = "" Then
    For i = 9 To 10
    Cells(i, 11) = ""
    Next i
ElseIf Cells(3, 10) = "" Then
    Cells(10, 11) = ""
Else: MsgBox "ÖLÇÜT SAYISI 2 İLE 8 ARASINDA OLMALIDIR"
End If

'hesaplamada kullanılan hücrelerin temizlemesi

For i = 3 To 10
    For j = 20 To 70
    Cells(i, j) = ""
    Next j
Next i

'ölçüt başlıklarının temizlenmesi
```

```
For i = 5 To 10
   If Cells(3, i) = "" Then
   Cells(2, i) = ""
   Cells(i, 2) = ""
   Else
   End If
Next i

'tanıtılan üçlü bulanık sayılar, açıklamaları ve kodlarının _
temizlenmesi

For i = 11 To 20
For j = 2 To 6
Cells(i, j) = ""
Next j
Next i

'boş bırakınız uyarısının temizlenmesi

For i = 3 To 10
For j = 3 To 10
If i >= j Then
Cells(i, j) = ""
End If
Next j
Next i

Cells(2, 11) = "ÖNEM DÜZEYLERİ"

UserForm1.Hide
End Sub
```

MS-Excel Macros for Matrix Calculations

```
Private Sub CommandButton1_Click()
Sheet4.Activate
Sheet4.Columns(4).Select
Selection.Insert Shift:=xlRight
Sheet4.Columns(3).Select
Selection.Copy
numRows = Selection.Rows.Count
numColumns = Selection.Columns.Count
Selection.Resize(numRows, numColumns + 1).Select
    Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:= _
    xlNone, SkipBlanks:=False
    Application.CutCopyMode = False
End Sub
-----------------------------
Private Sub CommandButton2_Click()
Dim sutun_say, sut_say, ABW, i, j As Integer
Cells(1, 2).Activate
sutunsay = 0
For i = 3 To 50
If Cells(1, i).Interior.ColorIndex = 40 And Cells(1, i).Interior.Pattern = xlSolid
Then
  GoTo 10
Else
sutun_say = sutun_say + 1
End If
 Next i
10
Sheet5.Activate
For i = 1 To sutun_say - 1
```

```
Sheet5.Rows(3).Select
Selection.Insert Shift:=xlDown
Next i
Sheet4.Activate
Sheet4.Range("c1").Activate
'Sheet3.Range("c1:v1").Select
 Selection.Resize(, sutun_say).Select
 Selection.Copy
Sheet5.Activate
Sheet5.Range("a2").Select
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:= _
        xlNone, SkipBlanks:=False, Transpose:=True
    Application.CutCopyMode = False
Sheet4.Activate
For i = 1 To 100
    If ActiveCell.Interior.ColorIndex <> 4 And ActiveCell.Interior.Pattern <>
xsolid Then
    ABW = i
   Cells(i, 1).Select
     Else
        GoTo 80
     End If
Next i
80
 Cells(ABW, 3).Select
 Selection.Resize(, sutun_say).Select
 Selection.Copy
 Sheet5.Activate
Sheet5.Range("b2").Select
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:= _
        xlNone, SkipBlanks:=True, Transpose:=True
                Application.CutCopyMode = False
```

```
MsgBox "process priorities has been defined, now start the next evaluation phase to define functional priorities", vbOKOnly
End Sub
------------------
Private Sub CommandButton3_Click()
Dim ABW, i, j As Integer
sut_say = 0
Cells(1, 1).Activate
j = 1
For i = 1 To 100
   If ActiveCell.Interior.ColorIndex <> 4 And ActiveCell.Interior.Pattern <> xsolid Then
   ABW = i
  Cells(i, j).Select
   Else
     GoTo 10
   End If
Next i
10
For j = 1 To 50
If Cells(ABW, j).Interior.ColorIndex = 4 And Cells(ABW, j).Interior.Pattern = xlSolid Then
  sut_say = sut_say + 1
Else
GoTo 20
End If
 Next j
20
 Selection.Resize(, sut_say).Select
 Selection.Copy
 Sheet9.Activate
Sheet9.Range("f3").Select
```

```
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:= _
        xlNone, SkipBlanks:=True, Transpose:=True
            Application.CutCopyMode = False
    Sheet9.Range("e1") = "Process  Priorities"
    Sheet4.Activate
    Cells(1, 3).Select
    sutunsay = 0
    For i = 3 To 50
    If Cells(1, i).Interior.ColorIndex = 40 And Cells(1, i).Interior.Pattern = xlSolid
Then
     GoTo 30
    Else
    sutun_say = sutun_say + 1
    End If
     Next i
    30
    Selection.Resize(, sutun_say).Select
     Selection.Copy
    Sheet9.Activate
    Sheet9.Range("E2").Select
        Selection.PasteSpecial Paste:=xlPasteValues, Operation:= _
        xlNone, SkipBlanks:=True, Transpose:=True

      Application.CutCopyMode = False
    Sheet4.Activate
    End Sub
    -----------------

    Private Sub CommandButton4_Click()
    Dim satir_say, sutun_say, i, j As Integer
    Sheet3.Activate
    sutun_say = 0
```

```
For i = 3 To 50
If Cells(1, i).Interior.ColorIndex = 40 And Cells(1, i).Interior.Pattern = xlSolid
Then
  GoTo 10
Else
sutun_say = sutun_say + 1
End If
 Next i

10
Sheet4.Activate

satir_say = sutun_say
Dim sari_say2 As Integer
sari_say2 = 1
For j = 1 To 50
If Cells(2, j).Interior.ColorIndex = 6 And Cells(2, j).Interior.Pattern = xlSolid
Then
  GoTo 70
Else
sari_say2 = sari_say2 + 1
End If
 Next j
70
Dim Sum_Absolute, Sum_Adjusted As Double
Sum_Absolute = 0
Sum_Adjusted = 0

For i = 2 To satir_say
   Cells(i, sari_say2 - 1) = Cells(i, 2) * (1 + Cells(i, sari_say2 - 2))
   Sum_Absolute = Sum_Absolute + Cells(i, sari_say2 - 1)
Next i
```

```
Cells(satir_say + 3, sari_say2 - 1) = Sum_Absolute

For i = 2 To satir_say
   Cells(i, sari_say2) = Round(Cells(i, sari_say2 - 1) / Sum_Absolute, 2)
   Sum_Adjusted = Sum_Adjusted + Cells(i, sari_say2)
Next i
Cells(satir_say + 3, sari_say2) = Sum_Adjusted
End Sub
---------------------------------
Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
If Target.Interior.ColorIndex = 34 And Target.Interior.Pattern = xlSolid Then
         UserForm14.Show
ElseIf Target.Interior.ColorIndex = 38 And Target.Interior.Pattern = xlSolid Then
         UserForm15.Show
ElseIf Target.Interior.ColorIndex = 39 And Target.Interior.Pattern = xlSolid Then
         UserForm16.Show
ElseIf Target.Interior.ColorIndex = 41 And Target.Interior.Pattern = xlSolid Then
         UserForm17.Show
End If
End Sub
------------------------------------
Private Sub CommandButton1_Click()
If OptionButton1.Value = True Then
ActiveCell.Value = Sheet2.Cells(5, 2)
ElseIf OptionButton2.Value = True Then
ActiveCell.Value = Sheet2.Cells(4, 2)
ElseIf OptionButton3.Value = True Then
ActiveCell.Value = Sheet2.Cells(3, 2)
ElseIf OptionButton4.Value = True Then
ActiveCell.Value = Sheet2.Cells(2, 2)

End If
```

```
UserForm14.Hide
End Sub
------------------------------------

Private Sub CommandButton1_Click()
If OptionButton1.Value = True Then
ActiveCell.Value = Sheet2.Cells(10, 2)
ElseIf OptionButton2.Value = True Then
ActiveCell.Value = Sheet2.Cells(9, 2)
ElseIf OptionButton3.Value = True Then
ActiveCell.Value = Sheet2.Cells(8, 2)
ElseIf OptionButton4.Value = True Then
ActiveCell.Value = Sheet2.Cells(7, 2)

End If
UserForm15.Hide
End Sub
-----------------------------------------

Private Sub CommandButton1_Click()
If OptionButton1.Value = True Then
ActiveCell.Value = Sheet2.Cells(12, 2)
ElseIf OptionButton2.Value = True Then
ActiveCell.Value = Sheet2.Cells(13, 2)
ElseIf OptionButton3.Value = True Then
ActiveCell.Value = Sheet2.Cells(14, 2)
ElseIf OptionButton4.Value = True Then
ActiveCell.Value = Sheet2.Cells(15, 2)

End If
UserForm16.Hide
End Sub
```

```
-----------------------------------------------
Private Sub CommandButton1_Click()
If OptionButton1.Value = True Then
ActiveCell.Value = Sheet2.Cells(17, 2)
ElseIf OptionButton2.Value = True Then
ActiveCell.Value = Sheet2.Cells(18, 2)
ElseIf OptionButton3.Value = True Then
ActiveCell.Value = Sheet2.Cells(19, 2)
End If
UserForm17.Hide
End Sub
----------------------------------------------
```

Correlations

```
Private Sub CommandButton1_Click()
Sheet9.Activate
If ComboBox1.Value = "Enterprise Goals" Then
   Sheet9.Range("a1:a100").Select
   Selection.Copy
End If
   If ComboBox1.Value = "Process Goals" Then
   Sheet9.Range("c1:c100").Select
   Selection.Copy
End If
   If ComboBox1.Value = "Processes" Then
   Sheet9.Range("e1:e100").Select
   Selection.Copy
End If
   If ComboBox1.Value = "Functional Characteristics" Then
   Sheet9.Range("g1:g100").Select
   Selection.Copy
```

```
End If
   If ComboBox1.Value = "Informational Characteristics" Then
   Sheet9.Range("i1:i100").Select
   Selection.Copy
End If
   If ComboBox1.Value = "Resource Characteristics" Then
   Sheet9.Range("k1:k100").Select
   Selection.Copy
End If
   If ComboBox1.Value = "Organizational Characteristics" Then
   Sheet9.Range("m1:m100").Select
   Selection.Copy
End If

Sheet11.Activate
'Sheet11.Cells.Clear
Sheet11.Range("a2").Select
      Selection.PasteSpecial Paste:=xlPasteValues, Operation:= _
      xlNone, SkipBlanks:=True, Transpose:=False

Sheet11.Range("b1").Select
      Selection.PasteSpecial Paste:=xlPasteValues, Operation:= _
      xlNone, SkipBlanks:=True, Transpose:=True
   Application.CutCopyMode = False
 UserForm18.Hide
MsgBox "This sheet is not saved at the end of the analysis,please keep a copy "
End Sub
----------------------------------------
Private Sub UserForm_Initialize()
ComboBox1.AddItem "Enterprise Goals"
ComboBox1.AddItem "Process Goals"
ComboBox1.AddItem "Processes"
```

```
ComboBox1.AddItem "Functional Characteristics"
ComboBox1.AddItem "Informational Characteristics"
ComboBox1.AddItem "Resource Characteristics"
ComboBox1.AddItem "Organizational Characteristics"
End Sub
-----------------------------------------------
Private Sub CommandButton1_Click()
bilgi = "Write Your Print Area"
   bilgi = bilgi + "Change the alues in the box according to your print area" +
Chr$(13) + Chr$(10)
   bilgi = bilgi + "Click OK for selection" + Chr$(13) + Chr$(10)
   bilgi = bilgi + "If you click CANCEL button, then whole table is printed" +
Chr$(13) + Chr$(10)
   düğme = "A1:K40"
   a$ = InputBox(bilgi, "PRINT RESULTS", düğme)
   ActiveSheet.PageSetup.PrintArea = a$
   ActiveWindow.SelectedSheets.PrintPreview
   Range("A13").Select
End Sub
---------------------------------------
Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)
If Target.Interior.ColorIndex = 44 And Target.Interior.Pattern = xlSolid Then
        UserForm19.Show
End If
End Sub
---------------------------------

Private Sub CommandButton1_Click()
If OptionButton1.Value = True Then
ActiveCell.Value = "SP"
ActiveCell.Font.ColorIndex = 42
ElseIf OptionButton2.Value = True Then
```

```
ActiveCell.Value = "WP"
ActiveCell.Font.ColorIndex = 5
ElseIf OptionButton3.Value = True Then
ActiveCell.Value = "NONE"
ElseIf OptionButton4.Value = True Then
ActiveCell.Value = "SN"
ActiveCell.Font.ColorIndex = 3
ElseIf OptionButton5.Value = True Then
ActiveCell.Value = "WN"
ActiveCell.Font.ColorIndex = 7
End If
UserForm19.Hide
End Sub
Private Sub UserForm_Click()
End Sub
```

Report Page

```
Private Sub CommandButton1_Click()
Dim i, j As Integer
For i = 1 To 50
  For j = 1 To 15
  Cells(i, j).NumberFormat = "0.000"
  If Cells(i, j).Value < 0.009 Then
          Cells(i, j).Font.Color = vbRed
     End If
   Next j
Next i

MsgBox "RED VALUES SHOWS THE ITEMS WHICH ARE TOO SMALL TO CONSIDER, IF SUCH A VALUE CREATES CONFLICTION THEN IT SHOULD BE REEVALUATED"
```

```
End Sub
------------------------------
Private Sub CommandButton2_Click()

'Sheet9.PrintPreview
 bilgi = "Write Your Print Area"
   bilgi = bilgi + "Change the alues in the box according to your print area" + Chr$(13) + Chr$(10)
   bilgi = bilgi + "Click OK for selection" + Chr$(13) + Chr$(10)
   bilgi = bilgi + "If you click CANCEL button, then whole table is printed" + Chr$(13) + Chr$(10)
   düğme = "A1:K40"
   a$ = InputBox(bilgi, "PRINT RESULTS", düğme)
   ActiveSheet.PageSetup.PrintArea = a$
   ActiveWindow.SelectedSheets.PrintPreview
   Range("A13").Select
End Sub
-----------------------
Private Sub CommandButton3_Click()
ActiveSheet.UsedRange.Select
Selection.Font.ColorIndex = 1
End Sub
```