

T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES

USER NEEDS ANALYSIS: A CASE STUDY FOR
SIMULATION BASED TRAINING IN TCDD

Dilek AĞIR

THESIS
FOR THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

SUPERVISOR
Asst. Prof. Dr. Serol BULKAN

ISTANBUL 2009

**T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES**

**USER NEEDS ANALYSIS: A CASE STUDY FOR
SIMULATION BASED TRAINING IN TCDD**

Dilek AĞIR

141100920060127

**THESIS
FOR THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING**

**SUPERVISOR
Asst. Prof. Dr. Serol BULKAN**

ISTANBUL 2009

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Asst. Prof. Dr. Serol BULKAN for his patience, inspiring guidance, constant encouragement and help throughout the period of my M.S. study and in the preparation of this thesis.

I would like to thank to my instructor and manager of TRENSIM Prof.Dr. Ercan ÖZTEMEL whom I received kind assistance in various part of preparing this thesis. I owe thank to him for the guidance, and dedication through this work.

Finally, I would like to thank to my family and classmates for their continual support during my studies.

Table of Contents

	PAGE NO
ACKNOWLEDGEMENTS.....	i
TABLE OF CONTENTS.....	ii
ÖZET.....	v
ABSTRACT	vii
ABBREVIATIONS	ix
LIST OF FIGURES	x
LIST OF TABLES	xi
CHAPTER I. INTRODUCTION	1
CHAPTER II. PREVIOUS STUDIES RELATED TO USER NEEDS ANALYSIS	4
II.1. INTRODUCTION.	4
II.1.1. User-Centered Design.....	4
II.1.2. Usability Engineering	5
II.1.3. Requirements Engineering.....	6
II.2. PREVIOUS STUDIES OF USER NEEDS ANALYSIS IN THE RELATED DISCIPLINES.....	9

CHAPTER III. UNA PROCESS AND METHODS	12
III.1 IDENTIFYING STAKEHOLDERS AND USERS	12
III.2 ELICITING USER NEEDS	14
III.2.1. Interviews	15
III.2.2. Questionnaires	18
III.2.3. Focus Groups.....	20
III.2.4. Field Observation	22
III.2.5. Verbal Protocol.....	24
III.2.6. Task Analysis	26
III.2.6.1. Workflow Analysis.....	27
III.2.6.2. Job Analysis	28
III.2.6.3. Combining Workflow Analysis / Job Analysis	28
III.2.6.4. Task Lists	29
III.2.6.5. Task Sequences	29
III.2.6.6. Task Hierarchies.....	30
III.2.6.7. Procedural Analysis.....	30
III.2.7. Contextual Design and Contextual Inquiry	31
III.3 PRESENTING USER NEEDS ANALYSIS FINDINGS	32
III.3.1. Requirements Documents.....	33
III.3.2. Scenarios	37
III.3.3. Use-Case Diagrams	39
III.3.4. Affinity Diagrams.....	43

III.3.5. Presenting Outcomes of Task Analysis'	44
III.3.5.1. Workflow Diagrams	44
III.3.5.2 Task Lists	45
III.3.5.3 Task Sequences	46
III.3.5.4 Task Hierarchies	46
III.3.5.5 User/Task Matrix	47
III.3.5.6 Detailed Task Descriptions from procedural analysis	47
III.3.5.7 Task Flowcharts	48
CHAPTER IV. A CASE STUDY IN TCDD	50
IV.1 AIM OF TRENSIM	50
IV.2 USER NEEDS ANALYSIS PROCESS OF TRENSIM	51
IV.2.1 Identifying Stakeholders and Users	55
IV.2.2 Eliciting User Needs	54
IV.2.31 Presenting User Needs Analysis Findings	60
CHAPTER V. CONCLUSION	72
REFERENCES	75

ÖZET

KULLANICI İSTERLERİ ANALİZİ: TCDD SIMULASYON TABANLI EĞİTİM PROJESİNDE VAKA ANALİZİ

Kullanıcı isterleri analizi, farklı uygulama alanlarının sistem ve ürün geliştirme süreçlerinde çeşitli teknikleri ve araçları kullanan çok disiplinli bir aktivitedir. Kullanıcı isterleri analizi, hem kullanıcı merkezli tasarım hem de ihtiyaç mühendisliği alanlarında önemini vurgulamaktadır. Kullanıcı isterlerinin analizi için her iki disiplin, kullanıcı merkezli tasarım ve ihtiyaç mühendisliği, kendi metot ve tekniklerini geliştirmiştir. Bu iki disipline ait çeşitli metotlar yayınlanmıştır. Fakat tüm bu metotlar kullanıcılara odaklanırken, kullanıcı isterleri analizi süresince hangi aktivitelerin kullanılması, analiz sonucunun neler içermesi gerektiği, sonuçların nasıl gösterilmesi ve iletilmesi konusunda farklı yaklaşımlar sunmaktadırlar.

“Sanal Ortam Tren Sürüş Eğitimleri İçin Milli Benzetim Sistemlerinin Geliştirilmesi” projesinin ana amacı tren makinistlerinin gerçek sürüş koşullarında eğitim alabilmelerini sağlayacak elektrikli lokomotif simulatörü geliştirmektir. Simulatörün ana kullanıcıları öğretmenlerdir ve bu nedenle kullanıcı isterleri analizi süreci onlarla birlikte yürütülmüştür. TRENSIM projesinin yapısına ve özelliğine uygun olarak, mülakatlar, bağlamsal sorgulama, düşüncelerini ifade etme ve iş analizi yöntemleri kullanılmıştır. Analiz sonuçları, ister dökümanları ve kullanım-durum diyagramaları olarak gösterilmiştir.

Kullanıcı isterlerinin elde edilmesine ve sonuçlarının gösterilmesine yönelik belirli tek bir yöntem yoktur. TRENSIM hem yazılım ve donanımsal ihtiyaçlara hem de kullanıcı ihtiyaçlarına eşit ağırlıkta önem verilmesi gereken bir projedir. Kullanıcı isterleri analizinde, Trensım yapısına uygun bilgi çıkarsama yöntemleri ve bunların sunulması Telelogic Doors ister yönetim aracı kullanılarak gerçekleştirilmiştir. Kullanıcılarla görüşmeler süresince bağlamsal sorgulama ve sesli düşünce ifade etme yöntemleri birleştirilmiştir. Sonuç olarak kullanıcı isterleri analizi TRENSIM

projesinin en önemli sürecidir. Proje ekibi zaman zaman yazılım sürecine ve donanımsal ihtiyaçlara daha fazla önem verse de bu sistemlerin ihtiyaçları doğrultusunda oluşturulması gerekliliğinin bilincinde isterlerin önemi ve eksiksizliği dikkate alınmıştır. Bunun sonucu olarak uygulama süreci sorunsuz bir şekilde yürütülebilir.

ABSTRACT

USER NEEDS ANALYSIS: A CASE STUDY FOR SIMULATION BASED TRAINING IN TCDD

User Needs Analysis (UNA) is a multi-disciplinary activity, deploying a variety of techniques and tools at different stages of development and for different kinds of application domain. User Needs Analysis emerges its importance both in User-Centered Design (UCD) activities which is related to Usability Engineering and Requirements Engineering (RE) activities. Both User Centered Design and Requirements Engineering have their own methods and techniques to analyze user needs for software engineering products. Several comprehensive User Centred Design methodologies and Requirements Engineering methodologies have been published, but they all focus on users, they disagree on exactly what activities should take place during the User Needs analysis, what end products of a User Needs Analysis should cover, how User Needs Analysis findings should be presented, and communicated. In this thesis we will represent the different User Needs Analysis activities on both UCD and RE methodologies. UCD and RE methodologies should be integrated, or at least aligned for an efficient analysis and the right product needed.

The main aim of the TRENSIM project is to develop an electrical locomotive simulator to train machinists using synthetic environment as in real life experiment. The main users of the simulator are instructors and the user needs analysis process is conducted with them to achieve the purpose and usability of the simulator. The techniques of eliciting user needs used in TRENSIM, regarding to its technical aspects and specifications, are field observation, think aloud protocol, contextual inquiry and task analyses. The finding of user needs analysis is represented in needs documents and as use-case diagrams.

There is not a particular way to analyse user needs and present its findings. TRENSIM needs both technical analysis and user analysis to achieve its aim. From the user needs analysis perspective, adequate techniques are chosen and findings well communicated and represented by using the requirements management tool called Telelogic Doors. Also the eliciting techniques of think aloud protocol and contextual inquiry are combined to get more efficient communication during interviews. As conclusion, the process of UNA is the most important phase of success of the TRENSIM project. The team members sometimes plunge in technical issues, they all consider the importance and completeness of the needs. As a result of this the implementation phase of the project has been executed seamlessly.

ABBREVIATIONS

CRI	: Customer Requirements Id
CRID	: Customer Requirements Identification Document
M	: Mandatory
O	: Optional
RE	: Requirements Engineering
SRI	: System Requirements Id
SRID	: System Requirements Identification Document
TRENSIM	: Sanal Ortam Tren Sürüş Eğitimleri İçin Milli Benzetim Sistemlerinin Geliştirilmesi Projesi
UCD	: User Centered Design
UCSD	: University of California San Diego
UNA	: User Needs Analysis
US	: Usability Engineering

FIGURES

	<u>PAGE NO</u>
Figure III.1 An actor in a use case diagram.....	40
Figure III.2 Use cases in a use case diagram.....	41
Figure III.3 A use case diagram depicting system boundary of the TRENSIM.....	42
Figure III.4 Workflow of a medical records application.....	45
Figure III.5 Symbol of start and end process.....	48
Figure III.6 Symbol of actions	48
Figure III.7 Symbol of decisions	49
Figure IV.1 Users in the simulation environment.....	54
Figure IV.2 User Needs Document in Doors.....	66
Figure IV.3 Use Case diagram of Scenario Editor.....	68
Figure IV.4 Use Case diagram of Malfunction Page.....	69
Figure IV.5 Use Case diagram of Rail Switch Control.....	70
Figure IV.6 Use Case diagram of Rail Switch Control in Enterprise Architect	71

TABLES

	<u>PAGE NO</u>
Table III.1 Problems and Solutions in Structuring Requirements	35
Table III.2 User/task matrix for using the Train Simulator	47
Table IV.1 Customer Requirements Table.....	56
Table IV.2 Software Requirements Identification Document.....	64

CHAPTER I

INTRODUCTION

Importance of the User Needs Analysis emerges both in development methodologies of User-Centered Design (UCD) activities and Requirements Engineering, as the most of the User Needs Analysis (UNA) activities are based on the software engineering products. Seen from the Requirements Engineering perspective, the User Needs Analysis, or at least part of it, is thus subsumed in Requirements Engineering activities. By contrast, the User-Centered Design community refers to User Needs Analysis with its rather broader focus on users, tasks, tools and environments as its special contribution to requirements capture (Lindgaard et al., 2006).

In the mid-1990s many large corporations with substantial IT branches responsible for in-house application development supplemented their internal development teams with outside vendors who brought their own development methodologies which, in some cases, were then adopted by the client organization (Lindgaard et al., 2006).

A large number of spectacular system failures (Sutcliffe, 2002) highlighted the importance of the requirements capture phase. A study by the Standish Group surveying a range of large projects (\$100,000+) in the United States, for example, found that 31% were cancelled before completion. Of the ones that were completed, 52.7% were 189% over budget on average at an additional cost of \$59 billion in the year the study was completed (Ewusi-Mensah, 1997). Problems included a lack of general agreement on project goals and objectives, insufficient platform experience in the project team, lack of senior management commitment, and poor project management (Ewusi-Mensah, 1997). However, perhaps most importantly the study found that, in general, project teams did not know how adequately to generate user information needs. Likewise, a European investigation of Software Engineering practices in industry found that ambiguity and changing requirements accounted for

many of the needs capture-related problems, as did a lack of trained manpower and inadequate methods (Lubars et al., 1993). The motivation of Requirements Engineering is thus to reduce the high cost of misunderstanding between users and designers so that computer systems are 'built to do what users want, on time and at reasonable cost' (Sutcliffe, 2002).

Therefore, it is very important to assess the goals and the needs of users in the early stage of design to achieve the purpose and usability of any system or product. Similarly it is important to determine the users' needs in their goals and tasks to provide an accurate training and usability of the simulator in simulation based training systems, as the aim of the simulation based training is to provide accurate training for train drivers regarding to the real life situations. Any inappropriate situation at any stage of the process might be based on the omitting the importance of the user need analysis stage. Because of that, definition of the user needs is a considerable step in the process.

From the education point of view, User Needs Analysis is a systematic effort for getting information, which are required for providing trainee's educational needs, in a right way. If a good concentration, methodology and analysis are provided, needs analysis can provide valuable data for defining the training needs and trainees' potential needs. A comprehensive need analysis should be in the center of the development process.

Simulation is the imitation of some real thing, state of affairs, or process. It is important to get right requirements to build a right simulation system or product to get the imitation as real as can be. Simulation is often used in the training of civilian and military personnel. In such situations they will spend time learning valuable lessons in a "safe" virtual environment. Often the convenience is to permit mistakes during training for a safety-critical system. Simulation systems allow individuals to train in highly realistic environments, provide immediate feedback to the trainee and present various levels of material to further one's skill level and expertise.

The train simulator, which is being developed in project of the TRENSIM, is used to train machinists on the ground. Professional technical training is very expensive in cost and labour force. Needs analysis is used for actualizing the goals of

professional technical training and improving this training in the simulated training system. It has instruction processes to deliver an all-round tool that realistically simulates potential critical situations which a machinist could be faced when driving the train. It permits a machinist to crash his simulated "train" without being hurt. Also it provides immediate feedback to the trainee. The main aim of TRENSIM project is to develop an electrical locomotive simulator to train the train drivers using synthetic environments for providing.

- Fundamental train driving capabilities (e.g. run, move, stop, etc.)
- Training in different weather and railway conditions
- The ability of preventing hazards and risks while driving
- Capabilities for monitoring and improving driver performance

This thesis aims are:

- Emphasising the importance of the user needs analysis
- Searching the approaches and techniques of the user needs analysis
- Searching the presenting methods of UNA results
- Illustrating the user need analysis phase of the TRENSIM as a case

To this end, this thesis is concerned with User Needs Analysis process. As the most of the user needs analysis activities are based on the software engineering products, it is subsumed in Requirements Engineering and User-Centered Design activities which are broadly described in chapter II. The first step of user needs analysis process is to identify the stakeholders and the users. Then the needs can be gathered directly from that identified users by using eliciting techniques. The gathered needs should be organized and presented for an agreement and communication between the users and the design team. In the chapter III, the way of presenting needs are described and illustrated. Finally, a case study is provided in chapter IV. Finally, chapter V summarizes our findings and discusses directions for future researches.

CHAPTER II

PREVIOUS STUDIES RELATED TO USER NEEDS ANALYSIS

II.1. INTRODUCTION

User Needs Analysis is a multi-disciplinary activity, deploying a variety of techniques and tools at different stages of development and for different kinds of application domains. User Needs Analysis emerges its importance both in User-Centered Design activities which is related to Usability Engineering and Requirements Engineering activities. Both User Centered Design and Requirements Engineering have their own methods and techniques to analyze the user needs for software engineering products. Several comprehensive User Centred Design methodologies and Requirements Engineering methodologies have been published, but they all focus on users, they disagree on exactly what activities should take place during the User Needs analysis, what end products of a User Needs Analysis should cover, how User Needs Analysis findings should be presented, and communicated (Lindgaard et al., 2006). In this thesis we will represent the different User Needs Analysis activities on both UCD and RE methodologies. UCD and RE methodologies should be integrated, or at least aligned for an efficient analysis and the right product needed.

II.1.1. User-Centered Design

The term ‘user-centered design’ originated in Donald Norman’s research laboratory at the University of California San Diego (UCSD) in the 1980s and became widely used after the publication of a co-authored book entitled: User-Centered System Design: New Perspectives on Human-Computer Interaction (Norman and Draper, 1986). Norman built further on the UCD concept in his seminal book The Psychology Of Everyday Things (Norman, 1988). In this book he

recognizes the needs and the interests of the user and focuses on the usability of the design (Abrás et al. 2004).

User-centered design is a design philosophy and a process in which the needs, wants, and limitations of the end user of an interface or document are given extensive attention at each stage of the design process (Almiral-Hill and Garreto-Domingo, 2007). In other words, User-Centered design, or UCD, is a comprehensive proprietary software development methodology driven by clearly specified, task-oriented business objectives and recognition of user needs, limitations, and preferences. Information collected using UCD analysis is scientifically applied in the design, testing, and implementation of products and services (Mauro, 2009).

The user-centered design tries to optimize the user interface around how people can, want, or need to work, rather than forcing the users to change how they work to accommodate the software developers approach.

Models of a user centered design process help software designers to fulfil the goal of a product engineered for their users. In these models, user requirements are considered right from the beginning and included into the whole product cycle. Their major characteristics are the active participation of real users, as well as an iteration of design solutions.

II.1.2. Usability Engineering

Usability engineering is a field that is concerned generally with human-computer interaction and specifically with making human-computer interfaces that have high usability or user friendliness. In effect, a user-friendly interface is one that allows users to effectively and efficiently accomplish the tasks for which it was designed and one that users rate positively on opinion or emotional scales. Usability Engineer is the person who is responsible for assessing the usability of an interface and recommending ways to improve it.

The term usability engineering implies more of a focus on assessing and making recommendations to improve usability than it does on design, though Usability Engineers may still engage in design to some extent, particularly design of

product prototypes. Usability engineers sometimes work to shape an interface such that it adheres to accepted operational definitions of user needs. While Usability Engineers are working on the usability of the product, they also concern about the user needs and recommend for the design which provides the user requirements.

Usability Engineers conduct usability evaluations of existing or proposed interfaces and their findings are fed back to the Designer for use in design or redesign.

II.1.3. Requirements Engineering

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting needs of the various stakeholders, such as beneficiaries or users. A systematic requirement analysis is also known as requirements engineering (Wieggers, 2003) It is sometimes referred to loosely by names such as requirements gathering, requirements capture, or requirements specification. The term requirements analysis can also be applied specifically to the analysis proper, as opposed to elicitation or documentation of the users' needs clearly and accurately, for instance (Laplante, 2007).

Zave (1997) describes that Requirements Engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families.

This definition is attractive for a number of reasons. First, it highlights the importance of “real-world goals” that motivate the development of a software system. These represent the ‘why’ as well as the ‘what’ of a system. Second, it refers to “precise specifications”. These provide the basis for analysing requirements, validating that they are indeed what stakeholders want, defining what designers have to build, and verifying that they have done so correctly upon delivery. Finally, the definition refers to specifications’ “evolution over time and across software

families”, emphasising the reality of a changing world and the need to reuse partial specifications, as engineers often do in other branches of engineering(Nusibeh, and Easterbrook, 2000).

The context in which RE takes place is usually a human activity system, and the problem owners are people. Engagement in an RE process presupposes that some new computer-based system could be useful, but such a system will change the activities that it supports. Therefore, RE needs to be sensitive to how people perceive and understand the world around them, how they interact, and how the sociology of the workplace affects their actions. RE draws on the cognitive and social sciences to provide both theoretical grounding and practical techniques for eliciting and modelling requirements.

These sciences which help the eliciting requirements are:

- **Cognitive psychology** provides an understanding of the difficulties people may have in describing their needs (Posner, 1993). For example, problem domain experts often have large amounts of tacit knowledge that is not amenable to introspection; hence their answers to questions posed by requirements analysts may not match their behaviour. Also, the requirements engineer may need to model users ’ understanding of software user interfaces, rather than relying solely on implementers’ preferences.
- **Anthropology** provides a methodological approach to observing human activities that helps to develop a richer understanding of how computer systems may help or hinder those activities (Goguen, and Jirotko, 1994). For example, the techniques of ethnomethodology (Goguen, and Linde, 1993).

have been applied in RE to develop observational techniques for analysing collaborative work and team interaction.

- **Sociology** provides an understanding of the political and cultural changes caused by computerisation. Introduction of a new computer system changes the nature of the work carried out within an organisation, may affect the structure and communication paths within that organisation, and may even change the original needs that it was built to satisfy (Lehman, 1980). A

requirements gathering exercise can therefore become politicised. Approaches to RE that address this issue include the “Scandinavian” approach, which aims to involve in the requirements definition process those most affected by the outcomes (Holtzblat, and Beyer, 1995).

- **Linguistics** is important because RE is largely about communication. Linguistic analyses have changed the way in which the English language is used in specifications, for instance to avoid ambiguity and to improve understandability. Tools from linguistics can also be used in requirements elicitation, for instance to analyse communication patterns within an organisation (Burg, 1997).

There is an important philosophical element in RE. RE is concerned with interpreting and understanding stakeholder terminology, concepts, viewpoints and goals. Hence, RE must concern itself with an understanding of beliefs of stakeholders (epistemology), the question of what is observable in the world (phenomenology), and the question of what can be agreed on as objectively true (ontology) (Nusibeh and Easterbrook, 2000). Such issues become important whenever one wishes to talk about validating requirements, especially where stakeholders may have divergent goals and incompatible belief systems.

RE activities of User Needs Analysis are:

- Identifying stakeholders
- Eliciting requirements
- Organizing requirements
- Presenting and communicating requirements
- Review and baseline requirements

II.2. PREVIOUS STUDIES OF USER NEEDS ANALYSIS IN RELATED DISCIPLINES

Some methodologies focus on User Needs Analysis data collection but have little to say about representation and communication of User Needs Analysis outcomes (Hackos and Redish, 1998), others are strong in all of these (Maguire, 1997, Beyer and Holtzblatt, 1998), but provide no clear way to update findings smoothly as new requirements emerge. There is thus a scarcity of appropriate support methodologies specifically designed to guide the iterative data collection, analysis.

Requirements Engineering attempts explicitly to integrate a user perspective into its processes. In terms of alignment of User Centred Design and Software Engineering, Beyer and Holtzblatt's (1998) Contextual Design methodology provides an entire user-centred process for generating design ideas and evolving a whole family of conceptual and practical models outlining work flow, the cultural setting in which the future application will be situated, and so forth. Taken together, these contextual design models yield a completely integrated picture of users, their organizational, cultural and other constraints which then inform the user interface design. The Contextual Design methodology treats the User Needs Analysis and user interface design processes as activities that appear to be independent of Software Engineering tasks. Thus, while it does provide an excellent means of situating User-Centred Design activities in a coherent framework, it makes it difficult to see how it could integrate smoothly with Software Engineering activities, especially where User-Centred Design and Software Engineering activities and goals overlap.

There are some distinct advantages to Beyer and Holtzblatt's approach: Requirements Engineering occupies an accepted role in systems development; User-Centred Design is struggling to attain recognition and acceptance in that arena (Vredenburg et al., 2002). A separate budget requiring a priori justification and post hoc accountability for User-Centred Design activities therefore reduces the risk of these being excluded from the project as well as securing their visibility.

The ‘usability context analysis’ method advocating that information about the context of use of a product should be collected by the usability team from the ‘people identified as having a good knowledge of the product, the users and their tasks’ (Thomas and Bevan, 1996). Thomas and Bevan describe a set of methods typically employed in the User Needs Analysis by User-Centred Design specialists, but they do not discuss how the results of these support the overall process of developing software. Hackos and Redish (1998) provide detailed instructions for conducting task analyses and interviews with users, but the link between these activities and Requirements Engineering tasks is unclear. Indeed, Sutcliffe (2002) clearly sees such activities as belonging squarely in the Requirements Engineering, whereas most User-Centred Design practitioners would probably regard precisely these activities as being central to the User-Centred Design tool kit.

From Constantine and Lockwood’s (1999) ‘Usage-centred design’ methodology perspective, use cases provide requirement engineers with a concise medium for modelling user requirements, while they provide user interface designers with a powerful task model for understanding user needs and guiding user interface design. For software engineers they see use cases being capable of guiding the design by effectively communicating objects to satisfy functional Requirements (Constantine and Lockwood, 2001). Like Beyer and Holtzblatt’s Contextual Design methodology, Usage-centred design is a model-driven process employing abstract models, namely user role, task, and content models. The user role model captures and organizes selected aspects of the relationship between users and the system. The task model represents the relationship between users and the system. The content model represents the goals of the user roles in the system, expressed in what Constantine and Lockwood refer to as ‘essential use cases’, which are ‘abstract, generalized and technology-free descriptions of the essence of a problem’ (Constantine and Lockwood, 2001). These essential use cases serve as the core model bridging user requirements and user interface design as well as connecting design and implementation. The use cases describe the user-computer interaction in concrete terms via a particular technology or a specific user interface (Constantine and Lockwood, 2001). The notion of use cases is widespread in software engineering (Jacobson, 1995), (Jacobson et al., 1999) but does not feature prominently in the HCI

literature, again with some exceptions (Constantine and Lockwood, 1999, Mayhew, 1999, Mayhew, 2003).

User needs Analysis method and techniques might be selected according to the users and stakeholders, the properties of product or system, and the goal of the analysis (Özbek, 2007). Although different processes are defined for user needs analysis in different studies, 3 steps are defined in most of the studies. These steps are given below.

- Identifying stakeholders and the users
- Eliciting the user needs
- Presenting the outcomes of the user needs

Accurate techniques and methods should be used to get successful results in the user needs analysis. Furthermore, determining the target stakeholders and users, not confusing requirement with needs, using more than one eliciting method, not confusing goals and tasks of the system, and presenting the data in a priority according to the importance are considerable points for getting successful results in the analysis.

However, the need still exists for researchers to develop much stronger methods in relationship with each others and contribute to the definition of tasks, activities and roles that, collectively, could improve both the design and development process and the outcome.

CHAPTER III

USER NEEDS ANALYSIS PROCESS AND METHODS

User Needs Analysis is a systematic method. It satisfies to set the needs for a specific community and topic in several domains and according to the results it suggests several changes for the corresponding needs. User Needs Analysis is a multi-disciplinary activity, deploying a variety of techniques and tools at different stages of development and for different kinds of application domains. All disciplines, related to user needs analysis, emphasis the importance of identifying who users are and involving the users in analysing phase for efficient, user friendly, exact product solutions needed. It is the process of defining and evaluating the needs.

User Needs Analysis Process can be defined under 3 main steps as listed below:

- Identifying stakeholders and the users
- Eliciting user needs
- Presenting User Needs Analysis findings

III.1. IDENTIFYING THE STAKEHOLDERS AND THE USERS

A user is someone involved in using a system when it is actually working (Abrams et al, 2004). Using does not only mean operating a system; testing, maintaining, and checking safety during system operation are all kind of activities of the users. A stakeholder is someone who has a justifiable claim to be allowed to influence the needs. In other words a stakeholder is anyone who could be materially affected by the implementation of a new system or application (Leffingwell and Widrig, 2003). All users are stakeholders. Other stakeholders may include:

- people who somehow affected by the system, such as clients and suppliers;

- managers who are concerned for the system to succeed, although they do not use it as such;
- regulators such as local and state governments and standards bodies, which are concerned about the effects the bl bsystem may have in its environment.

Effectively solving any complex problem typically involves satisfying the needs of a diverse group of stakeholders. Stakeholders will typically have varying perspectives on the problem and various needs that must be addressed by the solution.

Many stakeholders are users of the system, and their needs are easy to focus on because they will be directly involved with system definition and use it. However, some stakeholders are only indirect users of the system or are affected only by the business outcomes that the system influences. These stakeholders tend to be found elsewhere within the business, or in "the surrounds" of the particular application environment. In yet other cases, these stakeholders are even further removed from the application environment (Leffingwell and Widrig, 1999). From that point of view it is important to know where to find stakeholders, because identifying stakeholders is the first step of the User Needs Analysis.

An understanding of who these stakeholders are and their particular needs is an important factor in developing an effective solution of systems. Identifying the stakeholders may be a trivial or nontrivial step in problem analysis. The following questions can be helpful in this process (Leffingwell and Widrig, 2003).

- Who are the users of the system?
- Who is the customer (economic buyer) for the system?
- Who else will be affected by the outputs the system produces?
- Who will evaluate and approve the system when it is delivered and deployed?

- Are there any other internal or external users of the system whose needs must be addressed?
- Who will maintain the new system?
- Is there anyone else who cares?

After Identifying the stakeholders and user of the system, following steps should be followed (Galitz, 2002):

- Understand how people interact with the system.
- Understand the human characteristics important in design
- Identify the user's level of knowledge and experience.
- Identify the characteristics of the user's needs, tasks, and jobs.
- Who is the customer (economic buyer) for the system?

Planning users study in terms of how they define themselves in the work place(jobs, tasks, tools, and mental models), defining users in terms of their job titles, the tasks they perform, the tools they use, and the mental models of their workplace and tasks that they carry with them, provides users behavior in relation to the product. Also identifying users' knowledge and experience is another important issue. Giving users many new things to learn simultaneously causes significant job attrition, loss of productivity, proclivity for inventing inefficient and troublesome work around, and more (Hackos and Redish, 1998).

III.2. ELICITING USER NEEDS

Requirement elicitation is the process of discovering requirements for a system by communication with customers, system users, and others who have a stake in the system development (Sommerville and Sawyer, 1997).

Techniques for eliciting needs include interviewing users and other stakeholders, holding workshops, observing users at work, searching likely documents, and seeing the changes that users have made to existing systems.

Following are a set of recommended needs elicitation techniques.

- Interviews
- Questionnaires
- Focus Groups
- Field Observation
- Verbal Protocol
- Task Analysis
- Contextual Inquiry

These techniques can be used in combination. Their advantages are that they are effective in emerging the real needs for planned development efforts.

III.2.1. Interviews

An interview is a conversation between two or more people (the interviewer and the interviewee) where questions are asked by the interviewer to obtain information from the interviewee. Interview techniques are useful for identifying possible areas for more detailed analysis. Interviews are easy to conduct and direct, the unstructured interview can generate interesting points, statistical analysis can be run on the user's answers. Interview techniques are useful for investigating events which occur infrequently. The interviews can be recorded for a future analysis.

Interviewing users one-on-one is a quick way to gather a lot of information about their needs. It's generally not necessary to interview large numbers of people to get useful data as few as three or four interviews are often sufficient (Giacoppo, 2001).

On the other hand respondents are not committed to give correct answers and may often be influenced by what they believe the interviewer requires, or what they themselves wish to portray. The interviewer may need to acquire domain knowledge in order to know what questions to ask. There is a range of considerable bias due to the understanding by the users of the questions, and the subjective collected information might be misleading or inaccurate. The critical aspects are the choice of the place for the interview and how to conduct it (Bloodworth and Crew 2007).

There are several interview rules that can be used to get the most information from the people who are interviewed (Nielsen, 2002):

- Don't ask questions that can be answered with "yes" or "no."
- Don't ask leading questions.
- Don't use jargon.
- Don't draw attention to specific issues that you care about.
- Don't think emphasizing the product

The most important goal is to get as much feedback from users as possible without influencing their answers in any way. It's human nature to try to please an interviewer by giving the answers people think the interviewer would like to hear, so it's important not to indicate the feelings about the users' responses.

Some interviewees will talk for an hour about their issues and concerns without any prompting. Others will not have much to say unless they are asked prepared questions. In either case, interviewer should be mostly listening, prompting, and doing very little talking. While you usually want to keep users focused on issues that pertain to the system you are interviewing them about, don't feel pressured to complete all of your prepared questions. Allowing for open-ended explorations of their needs and ways of working will also yield good information. Sometimes you will find that information that doesn't seem pertinent at first will later give you valuable insights into the way users do their work and thus inform your design (Bloodworth and Crew 2007).

As you talk to users, you want to be looking for the following types of information from them:

- What tasks are they performing currently or would they like to perform?
- What would they like to see in a design?
- What are their expectations for the service?
- How can the current service be improved? (If possible, observe users using the current solution.)
- Any other specific data you need from users to inform your development effort.

There are three types of interviews as described below:

Structured: Structured interviews are useful in situations where the respondents' range of replies may be estimated and there is a need to clarify details, opinions or ideas. Structured interviews work well when the assessment goals are clear (Giacoppo, 2001).

Structured interviews require adherence to a very particular set of rules. Each question that is outlined should be read word for word by the researchers without any deviation from the protocol. Reactions to participant responses should be kept to a minimum or avoided entirely.

Structured interviews are the type used most often by quantitative researchers. The benefits are that it keeps the data concise and reduces researcher bias.

Semi-structured: Semi-structured interviews should only be carried out in a situation where broad issues may be understood, but the range of respondents' reactions to these issues is not known or suspected to be incomplete. This type of interview is mostly applicable in situations where both qualitative and quantitative feedbacks are required (Giacoppo, 2001).

Semi-structured interviews are most often used in qualitative studies. Data sets obtained using this style will larger than those with structured interviews.

Unstructured: An unstructured interview is when the interviewer does not impose any controls on the interview or the respondent. It is an intended to act as an exploratory conversation. Unstructured interviews are good for investigating potential emotional and / or sensitive personal issues (Giacoppo, 2001).

Unstructured interviews are best used when interviewers want to find as much information as possible about their topic. The benefit is that unstructured interviews often uncover information that would not have been exposed using structured or semi-structured interviews. Data sets collected using unstructured interviews will be larger than the other types.

III.2.2. Questionnaires

A questionnaire is a research instrument consisting of a series of questions and other prompts for the purpose of gathering information from respondents. A questionnaire is a series of questions asked to individuals to obtain statistically useful information about a given topic (Merriam, 2008). A questionnaire may bring out information from the respondent or it may start the respondent thinking or even doing some work on their own in order to supply the requested information. In any case, a questionnaire is a device that starts off a process of discovery in the respondent's mind (Kirakowski, 2000).

Usually, a questionnaire consists of a number of questions that the respondent has to answer in a set format. A distinction is made between open-ended and closed-ended questions. A closed-ended questionnaire is one that leaves no room for individual comments from the respondent. The respondent replies to a set of questions in terms of pre-set responses for each question. An open-ended questionnaire requests the respondent to reply to the questions in their own words, maybe even to suggest topics to which replies may be given (Kirakowski, 2000). The response options for a closed-ended question should be exhaustive and mutually

exclusive. Four types of response scales for closed-ended questions are distinguished (SurveyPedia, 2009):

- Dichotomous, where the respondent has two options
- Nominal-polychromous, where the respondent has more than two unordered options
- Ordinal-polychromous, where the respondent has more than two ordered options
- (bounded)Continuous, where the respondent is presented with a continuous scale

In general, questions should flow logically from one to the next. To achieve the best response rates, questions should flow from the least sensitive to the most sensitive, from the factual and behavioural to the attitudinal, and from the more general to the more specific.

Questionnaires have advantages over some other types of surveys in that they are cheap, do not require as much effort from the questioner as verbal or telephone surveys, and often have standardized answers that make it simple to compile data. When properly constructed and responsibly administered, questionnaires become a vital instrument by which statements can be made about specific groups or people or entire populations. Good questionnaire construction is critical to the success of a survey (Feucht and Lamp, 2006).

Questionnaires are also sharply limited by the fact that respondents must be able to read the questions and respond to them. Thus, for some demographic groups conducting a survey by questionnaire may not be practical. As a type of survey, questionnaires also have many of the same problems relating to question construction and wording. Inappropriate questions, incorrect ordering of questions, incorrect scaling, or bad questionnaire format can make the survey valueless (Feucht and Lamp, 2006).

III.2.3. Focus Groups

A focus group is a form of qualitative research in which a group of people are asked about their attitude towards a product, service, concept, advertisement, idea, or packaging. Questions are asked in an interactive group setting where participants are free to talk with other group members (Deek, 2008).

A typical focus group includes 8 to 12 people plus the facilitator. The facilitator follows at least loosely, a script or plan that has been decided in advance, laying out the issues to be covered or the questions to be addressed. The facilitator makes sure that no single person dominates the discussion, that the discussion stays close to the issues to be covered, and that all the issues in the plan are covered (Hackos and Redish, 1998).

Focus groups usually meet away from any work site and last about two hours. During a focus group session, the facilitator may ask questions about how users do things, what they have done in the past, and what they think they would do in certain situations. The facilitator may present ideas or demonstrate products and ask for immediate reactions. Focus groups are, therefore, a good way to get opinions, attitudes, preferences, and reactions from people. With focus groups, a group of people is observed in an interactive discussion. In the hands of skilled facilitator, the group interaction encourages a wide variety of ideas and opinions, with one participant influencing another to expand upon an idea or add a new concept (Hackos and Redish, 1998).

Skills and experience in conducting focus groups are important. The systematic ways of conducting focus groups are listed as below (USAID, 1996).

- **Selecting The Team:** Conducting a focus group interview requires a small team, with at least a facilitator to guide the discussion and a rapporteur to record it. The team should have substantive knowledge of the topic under discussion.
- **Select The Participants:** Types of groups and institutions that should be represented in the focus groups should be identified. This would be

determined by the information needs of the study. Then the most suitable people in each group should be chosen to consult several informants to minimize the biases of individual preferences.

- **Decide on Timing and Location:** Discussions last one to two hours and should be conducted in a convenient location with a some degree of privacy.
- **Prepare The Discussion Guide:** The discussion guide is an outline, prepared in advance, that covers the topics and issues to be discussed. It should contain few items, allowing some time and flexibility to pursue unanticipated but relevant issues. The guide provides that framework for the facilitator to explore, probe, and ask questions.
- **Conduct The Interview:** The facilitator should outline the purpose and format of the discussion at the beginning of the session, and set the group at ease. Participant should be told that the discussion is informal, and everyone is expected to participate.
- **Record The Discussion:** A rapporteur should record the discussion with tape recordings in conjunction with written notes. Discussion should be reported in participants' language, retaining their phrases and grammatical use.
- **Analyze Results:** After each session, the team should assemble the interview notes, the summaries, and relevant data to analyze trends and patterns.

This technique has several advantages. It is low cost and provides speedy results. Its flexible format allows the facilitator to explore unanticipated issues and encourages interaction among participants. In a group setting participants provide checks and balances, thus minimizing false or extreme views.

On the other hand, focus groups have several limitations. In most cases, focus groups do not examine users' behaviours with a product and are not held in the users' environments. Therefore, they are usually not a good way to understand how people

work or how they behave on the job. What people report about their work in a focus group and what may happen on the job may differ, significantly. Much of the work that people do is so automatic that they forget to mention it when just talking about it.

Focus groups often bypass users. In many cases, the participants are the supervisors, managers, and other decision makers, not the people who actually use the product.

Another limitation of focus groups is that some individuals will be more outspoken than others and are likely to influence the direction and conclusions of the group. Group dynamics may affect the users negatively and limit the variety of responses. An outspoken individual may dominate the conversation and lead the group to a consensus of opinion that individuals would not have reached on their own (Hackos and Redish, 1998).

III.2.4. Field Observation

Observational methods involve an investigator viewing users as they work and taking notes of the activity which takes place. Observation may be either direct, where the investigator is actually present during the task, or indirect, where the task is recorded on video-tape by the analysis team and viewed at a later time. The observer tries to be unobtrusive during the session and only poses questions if clarification is needed. Obtaining the cooperation of users is vital, so the interpersonal skills of the observer are important (Maguire, 2001).

According to Neuman (2000), field observation involves studying phenomena in context. Thus the researcher undertakes observation in the natural environment or location where people gather and events take place. This implies direct, face-to-face interaction with people in a natural setting. Thus, researchers go to people's homes or offices, street corners, restaurants and other places of social interaction. Researchers usually keep track of their observations in field notes. The greatest strength of direct observation is that there is virtually no time delay between the occurrence of an event and the recording thereof. Field observation is appropriate when the research

question involves learning about, understanding or describing the actions of an individual, or the behavior of or interactions between groups of people (Neuman 2000), (Barker and Angelopulo, 2005).

The systematic ways of conducting field observation are listed as below (Brown et al, 1998).

Selecting The Group: The research issues guide selection of a group to study. The researcher expects to invest a great deal of time and effort in the field setting, and an unfortunate choice of setting may lead to wasted time and resources.

Deciding What to Study: A researcher may want to study everything about a setting. Deciding what to study involves limiting the scope of the observations to behaviors that are relevant to central issues of the study.

Gaining Entry into The Group and Field: The involving into the group is the focus of the field observation may require the cooperation of members of the group.

Observing and Recording Data: During the field observation the tasks and the behavior of the group can be noted or recorded on video-tape. The notes include detailed information about the content of the study and the task as well as about the environmental issue affected the users and the task.

Organizing The Data and Presenting The Outcomes: It is the final step in conducting the field observation. However it may begin in the early stages of a field observation. This provides the opportunity to alter the need collection strategies if unpredictable situations occur. A typical strategy is to categorize and classify the tasks.

It is important that not only the UCD facilitators but also the designers and developers should go out into the field and have direct contact with the users. Field observations might give the designers some impressions and ideas that they would not have been able to obtain otherwise. After all, users are clients that the designers and developers are supposed to deliver products to (Gulliksen et al, 1998).

III.2.5. Verbal Protocol

Protocol analysis is a psychological research method that elicits verbal reports from research participants. Protocol analysis is used to study thinking in cognitive psychology (Crutcher, 1994), cognitive science (Simon and Kaplan, 1989), and behavior analysis (Austin and Delaney, 1998). It has found further application in the design of surveys and interviews (Sudman et al 1996).

Verbal protocols are written statements of what a user has said either during or after a task (Booth, 1989). Verbal protocol analysis provides a window on the user's mental model of what is going on while performing an activity by saying any and all thoughts that occur (Nemeth, 2004). Verbal protocols can provide much insight into the way the operator conceptualizes a task and the cognitive processes employed to perform tasks, analyzing verbal protocols can be difficult due to the time intensive and subjective nature of the analysis (Endsley et al, 2003).

In the broadest sense, verbal protocols are recordings of verbalizations. The recordings may be made on audiotapes or videotapes, as transcripts, or as notes taken by an observer. In human-computer interaction, verbal protocol methods are commonly used for two different purposes, understanding users' current work practices, and identifying usability problems in usability testing (Ericsson and Simon, 1993).

There are two type of protocol; concurrent and retrospective. A concurrent protocol is one that is taken while the user performs that task. A retrospective protocol is one that is taken once a task has been completed.

A concurrent verbal protocol has the advantage that it is likely to produce more data about a task and system. Distortions may arise in this type of protocol, however, as the act of providing the protocol may influence the user's behavior. A retrospective protocol has the advantage that it is less likely to affect the user's behavior at the task. Unfortunately, distortions in the protocol may occur as the user will only remember parts of the information that might have been provided by a concurrent protocol (Booth, 1989).

Lewis (1982) found that the method has both advantages and some limitations. The advantages of the method include:

- User hesitation and confusion can be used to identify what must be fixed.
- Users can reflect on what caused difficulty at a number of levels.

The method's limitations include (Nemeth, 2004):

- User may not be aware of all mental processes, or find it easy to describe.
- Talking aloud may cause task performance to take longer than normal.
- The detailed record of complicated behavior that the method produces can be difficult to summarize. Working with one participant at a time can be time-consuming.

Also there are three kind of verbalization (Jacko et al, 2003):

- Level 1: Vocalizations of thoughts that are already encoded in the verbal form (Talk Aloud).
- Level 2: Verbalization of a sequence of thought that are held in memory in some other form (Think Aloud).
- Level 3: Other verbalizations (retrospective report on thoughts not held in memory).

The Think-Aloud Method was developed by Clayton Lewis and is explained in Task-Centered User Interface Design Book (Lewis and Riemann, 1993, 1994). The method was further refined by (Ericsson and Simon, 1993).

Think aloud protocols involve participants thinking aloud as they are performing a set of specified tasks. Users are asked to say whatever they are looking at, thinking, doing, and feeling, as they go about their task. This enables observers to see first-hand the process of task completion rather than only its final product. Observers at such a test are asked to objectively take notes of everything that user's

say, without attempting to interpret their actions and words. Test sessions are often audio and video taped so that developers can go back and refer to what participants did, and how they reacted. The purpose of this method is to make explicit what is implicitly present in subjects who are able to perform a specific task (Kamandi and Habibi, 2008).

A related but slightly different data-gathering method is the talk-aloud protocol. This involves participants only describing their action but not giving explanations. This method is thought to be more objective in that participants merely report how they go about completing a task rather than interpreting or justifying their actions

III.2.6. Task Analysis

Task analysis analyses what a user is required to do in terms of actions and/or cognitive processes to achieve a task. A detailed task analysis can be conducted to understand the current system and the information flows within it. These information flows are important to the maintenance of the existing system and must be incorporated or substituted in any new system.

A task is what someone does to achieve a goal. As Donald Norman explains (Norman, 1988) “to get something done, you have to start with some notion of what is wanted-the goal that is to be achieved”. Also he says (Norman, 1988), that we cannot always articulate our goals clearly, but when we do tasks we are working towards goals.

Goals are a critical part of task analysis. To design successful products and designs you have to do task analysis on all levels from understanding the users’ goals to knowing the tasks they do to achieve those goals to understanding how they now carry out those tasks (Hackos and Redish, 1998).

A wide variety of different task analysis methods exist. TA methods can be used as an aid in the design stage of a new system, or the modification of an existing system. They can also be used as part of an audit of an existing system.

Different Types of Task Analysis Methods:

- Workflow Analysis
- Job Analysis
- Combining Workflow Analysis and Job Analysis
- Task Lists or Task Inventories
- Process Analysis, Task Sequences
- Task Hierarchies
- Procedural Analysis

III.2.6.1. Workflow Analysis

Workflow analysis is performed to understand how a group of users, accomplishes a particular process. It is a very important type of task analysis because the situation which different types of users are involved in the process is much more common than processes individuals do alone.

In a workflow analysis, the workflow should be extended to look not only at the users, who are involved in the actual work is wanted to be analysed, also at the stakeholders who are involved at the entry an exit points of the workflow. Workflow analysis shows that what at each step of the process. In other word, it allows investigators to identify which users are doing which tasks in a process. It is a horizontal picture of how work moves across users. It includes understanding the goals of all the users involved in the process. It includes understanding them as users, especially in relation to technology or other changes thought to be made to the workflow (Hackos and Redish, 1998).

Workflow Analysis defines user task processes. Workflow Diagrams document the steps for completing a task, show the user roles that perform particular steps, identify where the product is involved, and are used to evaluate product design (Henry, 2007).

III.2.6.2. Job Analysis

Job analysis is performed to understand the activities of a single user within a process. It is a vertical picture of all the activities that flow through a single user. A job analysis should involve direct observation of a user over time, or at very least, a logging our daily activities by a user.

The best way to do a job analysis is to spend time with users on the job. It includes observing and talking with the users as they do their jobs or have them keeping track of what they do. Just talking to them outside of the context of the work may not be enough. People do not remember what they do all day. Away from the situation, their sense of how much time they spend doing something or how often they do it may not match reality.

Job analysis can capture meaningful, quantitative data about how and where users spend their time. Information like frequency, time to complete, difficulty can all greatly help researchers designers and developers when it comes to developing, improving, discovering products for users.

III.2.6.3. Combining Workflow Analysis and Job Analysis

An overall task analysis might include both workflow analysis and job analysis. Workflow analysis help to find all the user groups involved and it helps to understand all the tasks that have to be included in the task list for the product. Job analysis help to understand all the work that a person in a certain position does during a day or week or month. If workflow analysis is a horizontal picture of how work moves across people, job analysis is a vertical picture of all the types of work that flow through a particular person.

A workflow and job analysis in tandem can provide a broad and deep picture of the users involved in a process, their tasks, and their interactions.

III.2.6.4. Task Lists

A task list is an inventory of all the tasks that users want to accomplish within the overall area that the product will handle. It is a list of what users have to be able to accomplish, not how to accomplish those tasks.

A useful task list is one that lists all the tasks that all the different types of users want to accomplish, and names the tasks from the users' point of view and in the users' words. Users' task list should drive decisions on the product's functionality. If the point of a product is to help people do works, the functionality of the product should come from understanding the overall work and the specific tasks that users need to accomplish-not just from the capabilities of the technology.

The techniques of getting a task list are observing and listening to users doing similar work, talking with users either individually or in groups about the work they do, and collecting scenarios of their work. As with job analysis, issues of frequency, criticality, time to complete, difficulty, and division of responsibility are relevant to gathering a task list.

III.2.6.5. Task Sequences

A task sequence is a series of tasks on the list that users must do or are likely to do in a certain order. Getting information on task sequences is a process analysis just like work analysis. The difference is that the information in work analysis is how a process is done that involves different users at different points in the process. For the type of task sequence, the information is how individual users each do a process. The information on task sequences is gotten with the same methods using to get task lists. The order of the action, in other words acting the task in specific steps, is considered for task sequences. The tasks which meet the goals are observed for the task lists.

Different users may do the same tasks in different order because of individual differences. In contrast different user doing the same task in the same way does not mean that the order is efficient. Users may be doing tasks in a certain order because;

- They have always done them that way.
- The current version of the product leads them to do it that way.
- Management has imposed that order to achieve consistency across users in the way that things are done
- Users have been trained to do the tasks in that order.

III.2.6.6. Task Hierarchies

A task hierarchy illustrates the tasks and subtasks of a process or activity it seeks to deconstruct and decompose tasks into its smallest components. It provides a picture of how many levels of tasks are involved in a process. It leads to decide which levels and how many levels of a process need to be investigated.

III.2.6.7. Procedural Analysis

When one specific task is divided into the steps and decisions that a user goes through in doing that task. The procedural level differs from the other types of task analysis because it includes working within the constraints of a specific part of the product (interface). It is very important to get a detailed picture of how users actually do the tasks at present (currently). Procedural analysis allows understanding both the physical steps that users take and the mental decisions they make while doing a specific task.

Procedural analyses of real users doing real work are often quite messy, but they are very informative. They show where current processes are difficult for users, thus providing opportunities to improve the process and to improve the software and documentation for doing the process.

III.2.7. Contextual Design and Contextual Inquiry

Contextual inquiry is a User-Centered Design method, part of the contextual design methodology. Contextual Design is a method which helps a cross-functional team come to agreement on what the users need and how to design a system for them (Holtzblatt, 2001).

Contextual Design approaches product design directly from an understanding of how users work. Great product ideas come from the combination of the detailed understanding of a user need with the in-depth understanding of technology(Beyer et al, 2004). Contextual Design starts with the recognition that any system embodies a way of working. Contextual Design is a method which helps a cross-functional team come to agreement on what their users need and how to design a system for them(Maunders et al, 2007).

Contextual Design has seven parts (Maunders et al, 2007):

1. Contextual inquiry uncovers who users really are and how they work on a day-to-day basis to understand the users: their needs, their desires and their approach to the work.
2. Work modelling capture the work of individuals and organizations in diagrams to provide different perspectives on how work is done.
3. Consolidation brings data from individual customer interviews together so the team can see common pattern and structure without losing individual variation.
4. Work redesign uses the consolidated data to drive conversations about how to improve work by using technology to support the new work practice.
5. The user environment design captures the floor plan of the new system. It shows each part of the system, how it supports the user's work, exactly what function is available in that part, and how the user gets to and from other parts of the system

6. Test with customers' paper prototyping develops rough mock-ups of the system using Post-its to represent windows, dialog boxes, buttons, and menus.
7. Putting it into practice of prioritization helps the transition to implementation by planning your system implementation over time. Object-oriented design helps you move from systems design to design of the implementation

Contextual inquiry involves collecting detailed information about user work practice by observing and interviewing the user while they actually work. The researcher should stay in the background and let the user lead the situation as much as possible. A contextual interviewer observes users as they work and asks about the users' actions step by step to understand their motivations and strategy. Through discussion, the interviewer and user develop a shared interpretation of the work (Holtzblatt, 2001). This means that researcher tries to form a partnership with user, learning (but not doing) as an apprentice while the user is the master of the work. This helps the researcher understand the user's work. The goal is to understand how and why something is done or why something is not done (Beyer and Holtzblatt, 1998).

Through contextual inquiry results, the team captures issues, draws work models, and develops a shared view of the user whose needs are being interpreted.

III.3. PRESENTING USER NEEDS ANALYSIS FINDINGS

The user needs, collected by eliciting techniques, usually start out as just that-a lot of facts needing to be synthesized and analyzed before they can lead to product design decisions. There are several methods to organize and present the needs. The way in which needs are documented and presented plays an important role in ensuring that they can be read, analysed, written, and validated. The completeness of the resulting product design depends on how the user needs and system requirements are expressed. Without some mechanism for prioritizing user needs, the translation process from needs into design may be difficult and lead to misunderstandings.

User needs can be presented in several ways:

- Documents
- Scenarios
- Use-Case Diagrams
- Affinity Diagrams
- Presenting Outcomes of Task Analysis

III.3.1. Requirements Documents

The result of user needs analysis is a documented agreement between users and developers about the product to be built (Hackos and Redish, 1998). The vision and scope document contains the business needs, and the user needs often are captured by eliciting techniques described in chapter two, section two. All these needs should be written in an organized and readable fashion to provide readable documents that the stakeholders and the product development team gets agreement on it.

The document should be written from the point of view of the user, in language and terminology that the user understands. It should avoid Engineering-speak, and should also avoid gratuitous design choices which dictate a particular solution (Stoney, 2008). User needs can be ranked according to importance, and later on in the project during the design phase it becomes apparent which needs will be met, and which will need to be re-negotiated due to feasibility, time or technical constraints.

The document is important not only because it is the vehicle by which the design team have their input into the product development process, but also because it is the basis of agreement amongst the user/customer and the project team.

A good structure helps organize the needs, making it easier to see what is necessary, where everything belongs, and how the different needs fit together. It also makes the needs easier to check for errors and missions (Kotonya and Sommerville, 1998)

The best structure for user needs follows the natural patterns of use, which is to say, the needs should be written down in the order that their users would come to them in their work. Sometimes different orderings are possible, and sometimes there are alternatives and special cases. Each of these situations represents a scenario, which at its most basic is a sequence of goals for the users.

The structure must give readers a way into the text. A good structure shows the needs at different levels of detail, and allows readers to focus on one section at a time. The users have to understand and feel that they own the user needs, whoever wrote them (Alexander and Stevens, 2002).

The aim is to show users just one simple structure, but to allow for any amount of detail. To do this, arrangement of what users need under chapter headings that represent the users' top-level goals is essential. Users can then see the whole pattern before diving into the details. There should be three levels of headings to organize the requirements: not to be used many more than that. Table III.I. lists some common problems that affect requirement structure, and the suggested solutions (Alexander and Stevens, 2002).

Table III.1: Problems and solutions in Structuring Requirements (Alexander and Stevens, 2002)

Problem	Solution
All readers need to understand the requirements	Write requirements in everyday language
Long lists of requirements are impossible to understand	Make a simple structure of chapters and sections to group the requirements
Requirements don't show what comes first	Organize the chapters and sections in time order so that they can be read as scenarios
Some requirements can be applied simultaneously, or in any order - a sequence is a needlessly tight ordering	Mark whether sections in the structure are sequences, parallels, or alternatives
The basic sequence of requirements doesn't show where what to do if something goes wrong	Add a section for each exception, at the place it could occur in the normal sequence
More than one sequence is possible in normal use	Describe all the sequences, mark them as alternatives
Some steps apply in several sequences	Describe them once, and reference them wherever needed
Some constraints apply to several requirements	If a constraint applies to a whole scenario, attach it to that section. If a constraint applies to separate sections, include cross-references
Users find it hard to get an overview of the whole document	1 All the above solutions 2 Add a simple informal diagram to support the text 3 Write an overview 4 Find out why they find it hard and restructure the document

A readable user needs document does not consist only of needs. Technical documents should have a context; there are constraints on what can be done and when; and there is an urgent need to keep track of need status, all the way through to completion.

Regarding to the European Space Agency's widely used Software Development Standards (Mazza et al, 1994), the structure of a user requirements document likes below;

1 General description

1.1 Product perspective

1.2 General capabilities

1.3 General constraints

1.4 User characteristics

1.5 Operational environment

1.6 Assumptions and dependencies

2 Specific requirements

2.1 Capabilities -the scenarios

2.2 Constraints - applying to the whole system

The vital thing is to get the users' needs across to the system designers. A Standard document can be extended or modified if the standard does not cover something needed. Traditionally, the needs are arranged in documents with structures that provide a hierarchy of headings with separate sections for items that are closely related. But the arrangement made it hard to find or understand all the needs related to the single desired result. To overcome this problem several tools are available.

If a good structure is created and agreed the user needs should fall into the place without difficulty. Needs are often stated badly initially and have to be worked on them to find out what is really wanted, and to be rewritten so that they are clear and precise (Alexsander and Stevens, 2002). Each user need should have a

- User type who benefits from the requirement;
- Defined desirable state for the user to reach, often an Object with a Qualifier;
- Mechanism to allow a test to be written against the requirement.

As an example of an imaginary requirement in traditional style ;

The call center operator shall be able to view details of a protected household within two seconds of issuing a query.

User type: The call center operator...

Result type (verb): ... shall be able to view ...

Object : ... details of a protected household...

Qualifier (adverbial phrase): ... within two seconds of issuing a query.

There is not a perfect set of applicable guidelines for good needs but some notations are widely acceptable to help to write better user needs as;

- **Use simple direct sentences:** Every need should be a single active sentence, as short as possible.
- **Use a limited vocabulary:** Write in a simple subset of language, avoiding terse that may confuse non-technical or foreign readers.
- **Identify the type of user who wants each need**
- **Focus on stating results:** Every need should name a single desired result.

Writing user needs is not a standalone activity, instead user needs must bridge the gap between the users of the product or system, and the development team who build the product. The document should be checked by the users and users feedbacks must be considered by re-organizing and re-writing the documents.

III.3.2. Scenarios

Scenarios are usually narratives that tell a story describing one or more tasks in a specific environmental situation (Cooper, 1999). Scenarios are stories about users and their activities. After eliciting user needs by task analyses, the key user tasks are often re-written as scenarios and/or stories which then can provide effective guides

for user interface design (Schuler and Namioca, 1993; Carroll, 1995; Ericsson, 1995). Scenarios are used for envisioning, clarifying and refining the needs and ideas for developing software systems and products. Its strengths lie in the elicitation and analysis of user needs encapsulated in ‘day in the life of’ style scenarios, and management of collaboration. Scenarios are critical both for designing the product and doing usability testing (Alexander and Maiden, 2004).

Scenarios come at different levels of detail.

- **Goal- or task-based scenarios** state only what the user wants to do. Scenarios at this level do not include any information on how the user goes about completing the scenario. This is also the type of scenario that you give users in a usability test. It gives them a reason and a goal for going to the site but it lets them show you how they would use the site to accomplish that goal.
- **Elaborated scenarios** include more details of the users' stories. Scenarios at this level give the development team a deeper understanding of the users. With that deeper understanding of what is important to users, of users' motivations and fears, and of users' ways of working, the team is more likely to develop the right content at the right level so that users find it comfortable and easy to work with.
- **Full-scale, task scenarios** that include the steps to accomplish the task. A full-scale scenario can either report all the steps that a specific user takes today to accomplish the task.

There are at least three limitations to the use of scenarios. One is that scenarios and stories rarely cover the entire user interface. These do not represent all possible user tasks in the new application (Lindgaard, 1994). The second limitation arises when a novel application is created. This includes situations when no organizational framework exists that can provide at least some of the necessary design constraints, or when users have not experienced the situations characterized. In such situations, a project team has little or no opportunity to verify its understanding of the problem space and its characterization of the users' tasks. Finally, the very concreteness of

scenarios and stories can present a hindrance when designing novel applications, especially if the application is intended to support flexible, open-ended tasks with multiple paths to the goal or in which the goal itself may change dynamically during the interaction (Lindgaard et al., 2006).

However, there is often no clear separation of the user and the system-side of the interaction in initial scenarios. The problem for the design of applications that support truly novel tasks is that, unlike the more mature scenario/story which usually presumes knowledge of the user's goal and a particular navigation path to accomplish that goal, such information is simply not available.

III.3.3. Use-Case Diagrams

Use cases emerged from the object-oriented development world. However, projects that follow any development approach can use them because the user doesn't care how the software is built. Use cases are at the center of the widely used Unified Software Development Process (Jacobson et al., 1999).

The objective of the use-case approach is to describe all tasks that users will need to perform with the system. In theory, the resulting set of use cases will encompass all the desired system functionality.

Use cases are intended to provide a high-level abstraction of tasks involving both the user and the computer (Constantine and Lockwood, 1999; Jacobson et al., 1992). A use case describes a sequence of interactions between a system and an external actor. An actor is a person, another software system, or a hardware device that interacts with the system to achieve a useful goal (Cockburn 2001). Another name for actor is user role, because actors are roles that the members of one or more user classes can perform with respect to the system (Constantine and Lockwood 1999).

Use-case diagrams provide a high-level visual representation of the user needs. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. A use case diagram is quite simple in nature and depicts two types of elements: one

representing the business roles and the other representing the business processes (Chitnis et. al, 2009).

Elements of use-case diagrams are actors,

Actors: Actor classes are used to model and represent roles for “users” of a system, including human users and other systems. The representation of an actor is shown in Figure III.1.They have the following characteristics (Si Alhir, 2000):

- Actors are external to a system
- Actors interact with the system.
- Actors may use the functionality provided by the system.
- Actors may provide functionality to the system.
- Actors may receive information provided by the system.
- Actors may provide information to the system.
- Actor classes have actors instances or objects that represent specific actors.

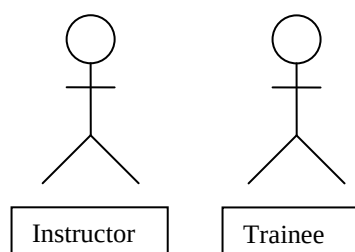


Figure III.1: An actor in a use case diagram

Use Cases: A use case is an external view of the system that represents some action the user might perform in order to complete a task (Fowler and Scott, 2000). Use cases are denoted as ellipses or ovals which are shown in Figure III.2. They may be enclosed by a system boundary or rectangle labelled with the name of the containing system. According to Sinan Si Alhir article they have the following characteristics (Si Alhir, 2000):

- Use cases are interactions or dialogs between a system and actors.
- Use cases are initiated by actors and may involve the participation of numerous other actors.
- Use cases may have extension points that define specific points within an interaction to which other use cases may be inserted.
- Use case classes have use case instances or objects called scenarios that represent specific interactions.

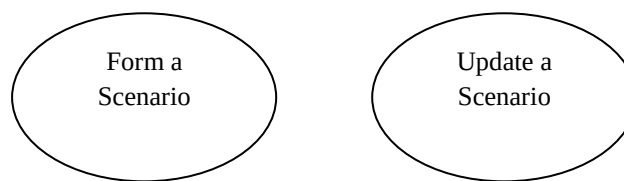


Figure III.2: Use cases in a use case diagram

System boundary: A system boundary defines the scope of what a system will be. A system cannot have infinite functionality. So, it follows that use cases also need to have definitive limits defined. A system boundary of a use case diagram defines the limits of the system. The system boundary is shown as a rectangle spanning all the use cases in the system. The system boundary is potentially the entire system as defined in the problem statement. But this is not always the case. For large and complex systems, each of the modules may be the system boundary (Chitnis et. al, 2009). An example of use case diagram in system boundary is shown Figure III.3.

Relationships: Association relationships between actor classes and use case classes are used to indicate that the actor classes participates and communicates with the system containing the use case classes.

Use cases share different kinds of relationships. A relationship between two use cases is basically a dependency between the two use cases. Defining a relationship between two use cases is the decision of the models of the use case diagram. Use case relationships can be one of the following:

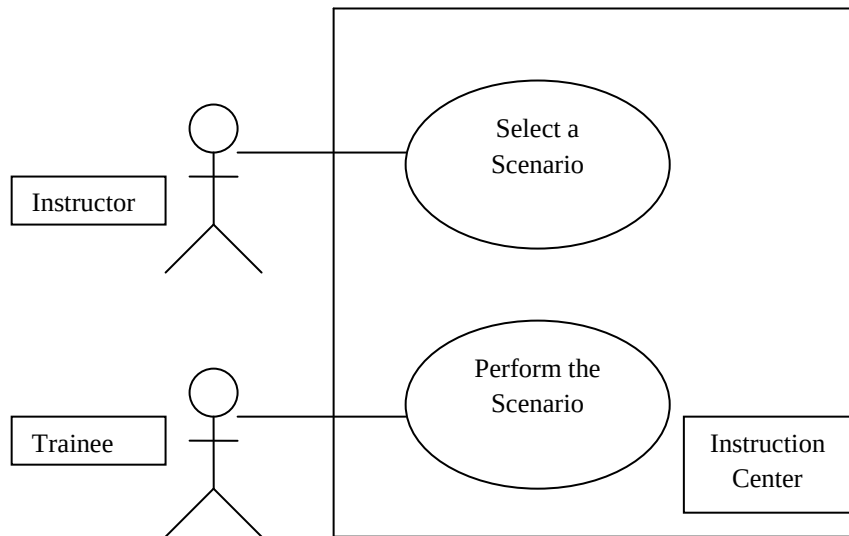


Figure III.3: A use case diagram depicting the system boundary of the TRENSIM

- **Include:** When a use case is depicted as using the functionality of another use case in a diagram, this relationship between the use cases is named as an include relationship. Literally speaking, in an include relationship; a use case includes the functionality described in the use case as a part of its business process flow.
- **Extend:** In an extend relationship between two use cases, the child use case adds to the existing functionality and characteristics of the parent use case. An extend relationship between use cases implies that the child use case enhances the functionality of the parent use case into a specialized functionality. The parent use case in an extend relationship cannot be replaced by the child use case.
- **Generalization:** A generalization relationship is also a parent-child relationship between use cases. The child use case in the generalization relationship has the underlying business process meaning, but is an enhancement of the parent use case. When you establish a generalization relationship between use cases, this implies that the parent use case can be replaced by the child use case without breaking the business flow

For drawing use case diagrams, any tool that supports use case diagrams can be used. A use case modelling tool provides a palette of options to draw actors and use cases and to define relationships between the use cases.

III.3.4. Affinity Diagrams

Another way to approach the analysis aspect of documenting the results of contextual observations/interviews that might work better when there is a larger number of observers/interviewers is known as constructing an “affinity diagram” (Mayhew, 1999). The affinity diagram is a business tool used to organize ideas and data. It allows large number of ideas to be sorted into groups for review and analysis.

Affinity diagramming starts by inferring design ideas from contextual interviews. It then proceeds to connect these ideas, subsume them under common headings and that way to build a hierarchy of ideas in a bottom-up manner (Holtzblatt and Jones, 1992). The technique is thus ideal for checking the completeness of the resulting user interface design, but it can be difficult to integrate new requirements once prototyping has begun. The requirement to create a new category once the number of ideas exceeds four in a given category can result in an entirely new arrangement of categories that is no longer consistent with the original conceptualization of the emerging User Interface design. The need to start again can conflict with stringent development deadlines (Lingaard et al, 2006).

In this approach, each observer/interviewer re-records each individual observation or piece of data from his or her own set of notes onto a separate self-sticking note. Then all observers/ interviewers meet and as a group put their respective notes up on a big blank wall one at a time, clustering notes together that seem to be related. As more and more notes go up, the emerging structure of clusters is rearranged in ways the group agrees upon. In this way, a final structure of distinct groups of issues and observations evolves. Then the team members label clusters and group them into higher-level categories. Thus, all the rich data gathered across a large number of contextual observations/interviews by a large number of individuals can be exhaustively documented and shared as well as organized and reduced to their commonalities (Mayhew, 1999).

III.3.5. Presenting Outcomes of Task Analysis

To describe the tasks that have been observed and discussed during site visits, seven methods will be expressed:

- Workflow diagrams
- Task lists
- Task sequences
- Task hierarchies
- User/Task matrices
- Detailed task descriptions from procedural analysis
- Task flowcharts

III.3.5.1. Workflow diagrams

Workflow diagram is the outcome of the workflow analysis. Workflow diagrams help to illustrate a collaborative sequential task or set of tasks. Tasks of this type usually move through a number of steps. Information is often passed from person to person to meet the overall goal of the task. In other words, the workflow diagram shows how tasks proceed from one group of users to another. It should also show how objects are altered as they are moved along.

According to Hackos and Redish (1998), a workflow diagram should only include people who perform the task and what they do. Figure III.4 illustrates a workflow example of a medical records application.

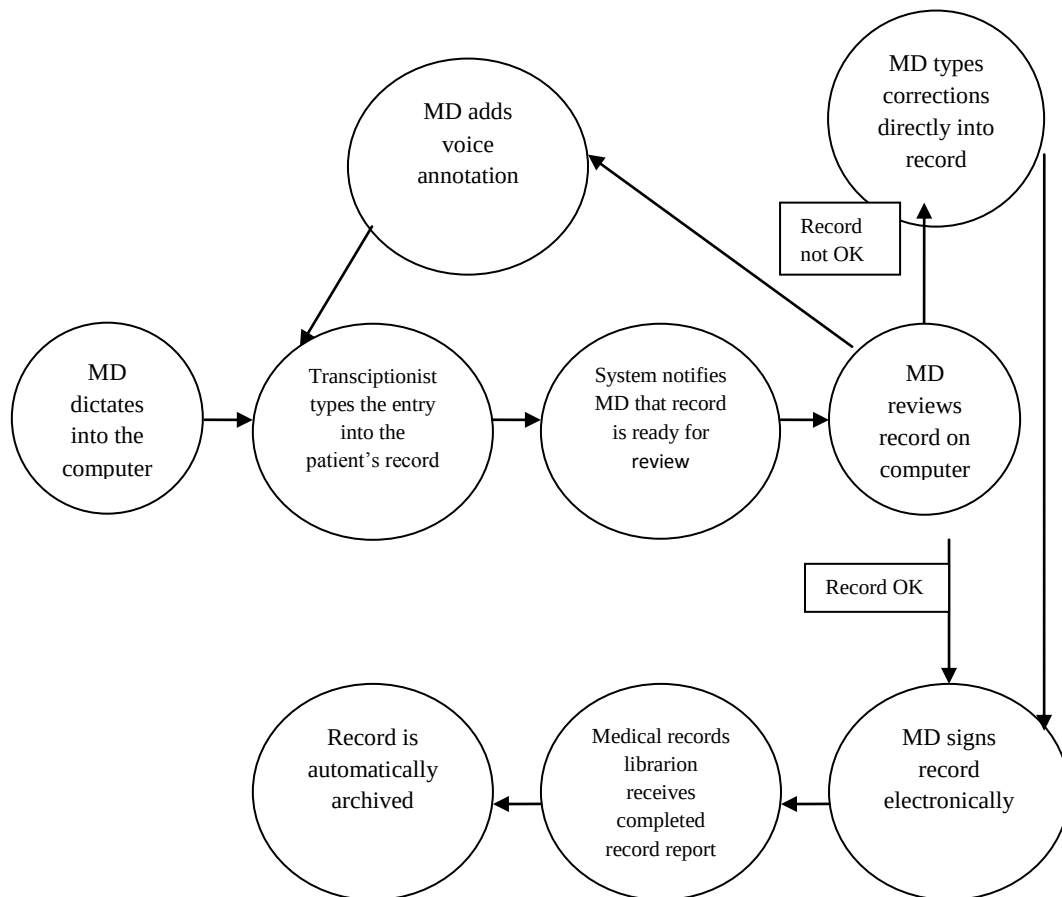


Figure III.4: Workflow of a medical records application (Hackos and Redish, 1998)

III.3.5.2. Task lists

A task list is one of the outcomes of user needs task analysis. A task list should be prepared for each category of user. It lists all of the tasks that must be performed by a user in a given position. Effective task lists begin each task with a verb that describes what the employee must do. The list states all the specific duties that the employee must perform even if the task is not one that is performed frequently.

A useful task list is one that lists all the tasks that all the different types of users want to accomplish, and names the tasks from the users' point of view and in the users' words. Users' task list should drive decisions on the product's functionality. If the point of a product is to help people do work, the functionality of the product should come from understanding the overall work and the specific tasks that users need to accomplish-not just from the capabilities of the technology (Hackos and Redish, 1998).

Partial task list for an e-mail program.

- Write a message.
- Send a message.
- Receive a message.
- Read message that you received.
- Reply to a message.
- Save a message to look at it later.
- Send a formatted file with the message.
- Send the same message to several people.
- Keep an address book.

III.3.5.3. Task sequences

Task sequences are often ordered lists of the sequence in which the tasks are most often performed. Creating a task sequence in which the tasks are most often performed and in which tasks are performed first and last to reach the goals.

The task sequence can be illustrated as an ordered list. This list is different from the collaborative workflow diagram because a task sequence is performed by a single user.

III.3.5.4. Task hierarchies

The interrelationships among tasks can be clarified by placing them in a hierarchical relationship to one another. Tasks at the top of the hierarchy point more closely to the goals the user is trying to achieve, while tasks lower in the hierarchy show what activities are required to achieve the goals.

A task hierarchy can be developed at a high level to analyze user goals or can be extended into lower levels to identify subtasks. Task hierarchies can also be used to show many levels of tasks in very large systems.

III.3.5.5. User/Task matrix

It is useful to compare lists of tasks and task sequences to the user groups have been identified. A user/task matrix illustrates how user groups relate to task performance. Some user groups never perform a particular task while others perform it frequently. The user/task matrix makes all relationships clearer (Hackos and Redish, 1998). An example of User/task matrix table is shown in Table III.2.

Table III.2: User/task matrix for Using of The Train Simulator

Tasks	Instructor	Trainee	Technician
Form a Scenario	Frequent	Never	Sometimes
Terminate a Scenario	Frequent	Never	Rare
Perform a Scenario	Sometimes	Frequent	Never
Insert a Comment about Training	Frequent	Never	Never
Insert Parameters in a Running Scenaio	Frequent	Never	Never
Drive The Simulated Train	Sometimes	Frequent	Rare

III.3.5.6. Detailed task descriptions from procedural analysis

Amount of the detail of tasks is need depends on the nature of the task and how much the users tasks going to be changed. Knowing how the task is current done by users gives important insights into how the users think about their tasks. A detailed task description might include a step-by-step account of the actions performed and decisions made by the user.

III.3.5.7. Task flowcharts

Some tasks proceed through a series of decisions, actions, and alternative paths depending on unique conditions. The users often have a variety of optional way of doing the same thing. A flowchart shows the detailed steps of tasks that show the branches for the conditional decisions and alternative paths. It makes the options obvious and shows the steps and decisions visually. Also flowcharts are simple diagrams that map out a process so that it can easily be communicated to other people.

Most flow charts are made up of three main types of symbol:

- Elongated circles, which signify the start or end of a process is shown Figure III.5.

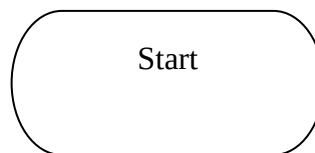


Figure III.5: Symbol of start and end process

- Rectangles, which show instructions or actions, are shown in Figure III.6.

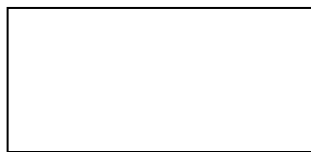


Figure III.6: Illustration of actions

- Diamonds, which show decisions that must be made, is shown in Figure III.8.

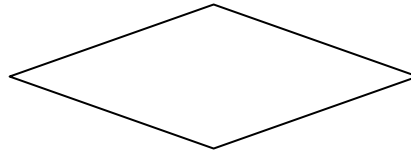


Figure III.7: Symbol of decisions

Symbols are connected one to the other by arrows, showing the flow of the process.

Task flowcharts and workflow diagrams are useful for communicating the details of a task to developer familiar with using flowcharts in their own work.

CHAPTER IV

A CASE STUDY IN TCDD

IV.1. AIM OF TRENSIM

There is a need for fully capable simulators to train more realistically the machinists in a cost and time effective manner. In Turkey, there are 3 train simulators and 2 of them possess the old technology. It is estimated that there is a need for about 30 simulators. The simulators would either be procured or would have to be developed nationally. One of the main problems with procurement is not being able to access the source codes for possible update and modifications towards future needs which is otherwise too costly and limit the possible benefits. Even the source codes are become available; it becomes too difficult for the national developers to understand and manage them which in turn require continuous support of the original developers. This obviously necessitates national research and development based simulator developments (TRENSIM, 2007).

The main aim of TRENSIM project is to develop an electrical locomotive simulator to train the machinists using synthetic environment for providing

- Fundamental train driving capabilities (e.g. run, move, stop, etc.)
- training in different weather and railway conditions
- the ability of preventing hazards and risks while driving
- capabilities for monitoring and improving driver performance

through research and development studies using enhanced simulation technologies. The main output of this project will be E43000 type electrical locomotive simulator which is designed according to Turkish National Railway (TCDD) needs.

The simulator which will be developed will have train cockpit and related software including related modelling capabilities databases, map processing, mission planning, performance evaluation, debriefing, distance based learning and automatic curriculum preparation, automated fault diagnosis. There will be mathematical models for simulated cockpit components. Artificial Intelligence technologies will be utilised especially for performance evaluation of the drivers. An e-learning environment will be developed according to the users needs. A motion platform which increases the fidelity of the training will be developed. Waterfall approach will be used in software development and quality standards will fully be implemented during project progress (TRENSIM, 2007)

Related experience and knowledge gained through this project would be used to create a national capability in train simulation studies which could also be utilized in other domains.

The results of this project will be used to create national simulators which could also be updated and modified towards contemporary needs of Turkish National Railway Association. It was believed that creating simulators nationally through research and development will create cost savings and improve national technology levels as well as create new employment areas.

The project proposal includes possible benefits, advantages and other details of the project such as project phases, work packages, and project organization and management principles.

IV.2.USER NEEDS ANALYSIS PROCESS OF TRENSIM

The scope of this thesis is set to cover only the user needs of the development; system and subsystem's technical specification is not discussed in any detail. The project is divided eight main modules. These modules are listed below.

- Hardware of the Platform
- Dynamic Modelling of the System
- Simulation Management System

- 3D Image Simulation System
- Voice Simulation System
- Movement Simulation System
- Performance Evaluation System
- Distance Based Learning

In this thesis the requirements and examples are given for the user needs of the Simulation Management System. The components of the Simulation Management Systems (TRENSIM, 2008)

- Scenario Management System
 - Scenario Preparation
 - Scenario Running
 - Mulfunctions
- Train, locomotive, and railway car attributes of E43000
- Monitors in Control and Watching Station
- Debriefing

Control and Watching Station is the center of the TRENSIM system control, and trainees' performance tracking, and control. There are 5 monitors. System management and information flow will be hold in this station. The instructor will be able to setup scenarios, training plan, and car park, which contains locomotive, train cars and trains.

Firstly the stakeholders and the users are identified, and then the needs gathered by interviewing with users. Then the user needs analysis outcomes presented as requirements document using requirements management tool, Telelogic Doors, and as use-case diagrams using Enterprise Architect.

IV.2.1. Identifying stakeholders and users

Instructors are responsible to preparing and starting scenarios for machinists training, tracking and intervene in the scenarios while the trainee is running the simulator. So the instructors can be use the system effectively, regarding to the right training scenarios and simulation usability needs.

Firstly we focus on the identifying the stakeholders and the user of the Train Simulator. The stakeholders, shown in Figure IV.1, are trainees (machinists), Instructors (operators), Technicians, system administrators, distance learning trainees, and manager of the Eskişehir Education and Training center of TCDD. The users of the simulator are trainees (machinists), instructors (operators), technicians, system administrators. Trainees are secondary users who actually do not know what is needed for training purpose on the simulator. As the aim is to train machinist in synthetic environment as in real life experiment, the instructors are our main users. The system should be designed as real train locomotive and railway systems and traffics, and geographical conditions.

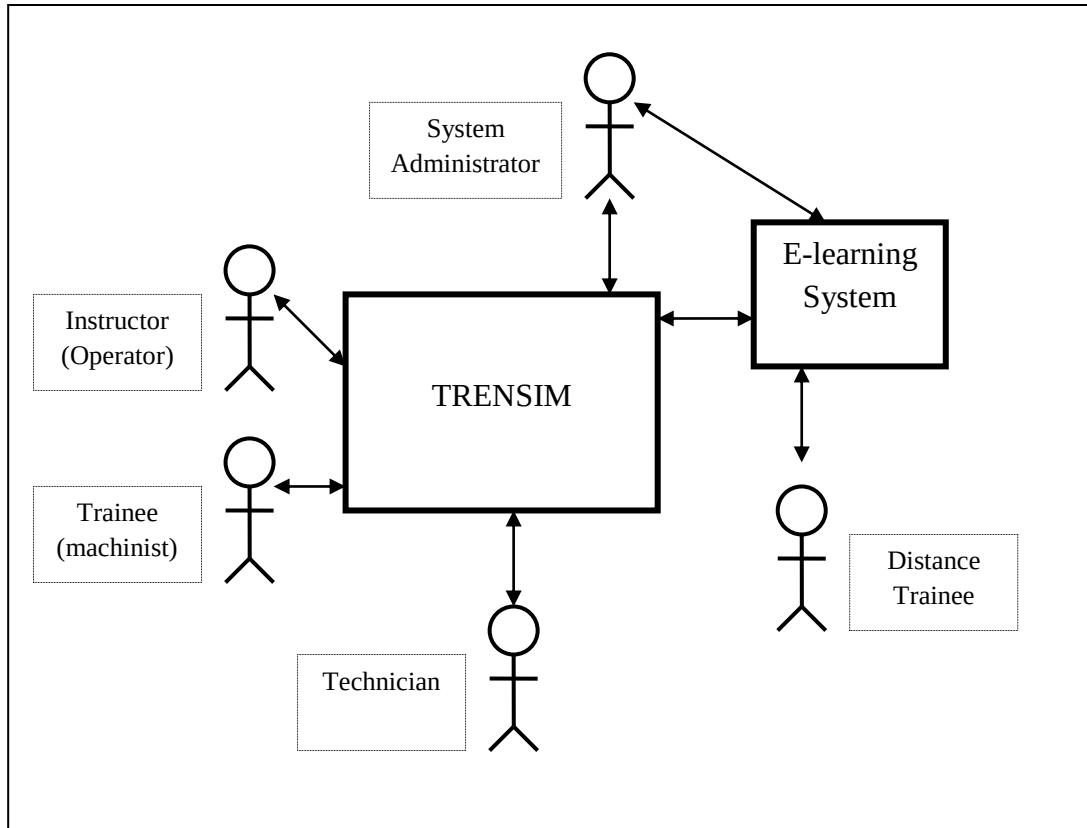


Figure IV.1: Users in the simulation environment (TRENSIM, 2008)

IV.2.2. Eliciting user needs

The second step is eliciting user and system needs. For this purpose we visited the TCDD Eskişehir Education and Training Center and our techniques to get user needs were;

- Field Observation
- Think Aloud Protocol
- Contextual Inquiry
- Task Analysis
- Trained as a Trainee
- Examine Technical Documents

On the first visit the analysis and design team had general railway and train education in class environment. Also the two old system simulators were visited. This education provided general ideas to researchers and design team about railway system and driving train. Also technical documents, about railway traffic system, dynamics of the E43000 locomotive and railway cars, were collected.

Secondly, we conducted field observation with instructor on subject to how the existing simulators are designed, what the users goals are, and what their tasks they do to achieve the goals. To get something done, we started with some notion of what is wanted the goal that is to be achieved. It was important to understand current simulation system and the information flows for designing new Simulator. We listed the goals and tasks of the scenario management and control and watching station where training controlled and viewed, for the next interviews. The primary needs for module of the simulation management system are listed 4 main components. One of the main components of the customer requirements, The scenario management system, is listed in Table IV.1 as customer requirements.

The Customer Requirement Identification (CRI) provides the tracibility of the needs. The second column express the text is a need or not. True means that it is a need. False means that it is not a need. The need text column explain the need. In the need state column it can be traced whether it is mandatory (M) or optional (O). The project is divided 8 packages. Package in charge column identify the package of the needs. The scenario management system is one of the components of the simulation management system. The last column shows who is responsible for the last modification of the need.

Table IV.1: Customer Requirements Table

Customer Requirement Id	Need ?	Need Text	Need State	Package in charge	Who modified
	False	1.1 SCENARIO MANAGEMENT			
CRI 09	True	There should be 5 monitors in Control Station and 4 monitors in Watching Station.	M	Simulation Management System	Admin
CRI 10	True	Scenario packages could be formed for training and examination.	M	Simulation Management System	Admin
CRI 11	True	The scenarios in the system could be listed and modified.	M	Simulation Management System	Admin
CRI 12	True	A car park could be constructed with types of locomotives and train cars.	M	Simulation Management System	Admin
CRI 13	True	The environment and climate condition could be changed both while forming and running scenarios.	M	Simulation Management System	Admin
CRI 14	True	The Objects (crow, car etc.) could be located on railway route While the training scenarios running	M	Simulation Management System	Admin
CRI 15	True	The same instructors' monitors should be both in the control station and in the watching station.	M	Simulation Management System	Admin
CRI 16	True	The risk situations should be set up by sending malfunctions to the system at the	M	Simulation Management System	Admin
CRI 17	True	There should be triggers to perform some events like object positioning, weather condition changing, signal changing, etc.	M	Simulation Management System	Admin
CRI 18	True	The train acts, train and railway route attributions should be monitored in one of the instructors monitors.	M	Simulation Management System	Admin
CRI 19	True	These should be monitored as numerical and pictorial in Monitor 2	M	Simulation Management	Admin

Customer Requirement Id	Need ?	Need Text	Need State	Package in charge	Who modified
		• All indicators on the control platform • Manometers on the control platform • Control states on the control platform		System	
CRI 20	True	Digital CGI route video should be monitored in Monitor 3.	M	Simulation Management System	Admin
CRI 21	True	The Platform should be monitored as video record in Monitor 4.	M	Simulation Management System	Admin
CRI 22	True	The cabin malfunction display should be monitored in Monitor 5.	M	Simulation Management System	Admin
CRI 23	True	When indicators or/and manometers exceed limits, a warning sign should be given in monitor.	M	Simulation Management System	Admin
CRI 24	True	The parameters of locomotive should be adjustable.	M	Simulation Management System	Admin
CRI 25	True	The parameters of train car should be adjustable.	M	Simulation Management System	Admin
CRI 26	True	There should be a car library that includes train cars and their equipments' specifications. The data in the library could be updated.	M	Simulation Management System	Admin
CRI 27	True	New train cars could be added the car library.	M	Simulation Management System	Admin
CRI 28	True	The modification of the car library should be send to related simulations automatically.	M	Simulation Management System	Admin
CRI 29	True	Different train car types should be defined for simulated train.	M	Simulation Management System	Admin
CRI 30	True	The simulated train could include 100 train cars and 2 locomotives. The locomotives could be plant on front of the train and end of the train.	M	Simulation Management System	Admin

Customer Requirement Id	Need ?	Need Text	Need State	Package in charge	Who modified
CRI 31	True	The Train parameters should be adjustable.	M	Simulation Management System	Admin
CRI 32	True	The pressure and tension between train cars should be monitored in real time.	M	Simulation Management System	Admin
CRI 33	True	Scenarios should be stopped and carry on the same point.	M	Simulation Management System	Admin
CRI 34	True	Simulated train should be positioned on any point on the railway route.	M	Simulation Management System	Admin
CRI 35	True	While forming a scenario, the railway route should be set.	M	Simulation Management System	Admin
CRI 36	True	The state of rail switches could be organized when forming or running scenarios	M	Simulation Management System	Admin
CRI 37	True	The state of signals could be organized when forming and running scenarios.	M	Simulation Management System	Admin
CRI 38	True	The adhesion level could be set on beginning of the simulation.	M	Simulation Management System	Admin
CRI 39	True	Virtual trains could be positioned to scenarios.	M	Simulation Management System	Admin
CRI 40	True	Data of system and training could be back up.	M	Simulation Management System	Admin
CRI 41	True	Run-for could be set while training scenarios running.	M	Simulation Management System	Admin

Then it was decided to use two techniques to gather detailed needs. When the field observation was conducted with the instructors about how they train the trainees in synthetic environment, the techniques of thinking aloud protocol and contextual inquiry were combined. Although observers do not attempt to interpret user action and words, during the thinking aloud protocol, discussion on actions with the instructor was held as if the contextual inquiry approach was followed.

The team observed an instructor while he preparing a scenario and also when he running a scenario, in his workplace that control station. There are 5 monitor in the control station. All these monitors are analysed. While he were preparing a scenario, the team asked him to say whatever he were looking at, thinking, doing, and feeling, as he goes about his task. This were enabled us to see first-hand the process of preparing a scenario. While he was performing his task, the team took notes and videotaped him so the team could go back and refer to what he did. Furthermore, the team interrupts and asked their questions and wanted to interpret some actions step by step, while the instructor preparing the scenario. However it was time consuming, it was a necessary to carry out such a complicated task. This helped us to understand the users work. Through discussion, the team and user developed a shared interpretation of the work.

On the Watching Station there are 4 monitors for tracking the trainee actions and performance, and simulation response. We also observed what the instructor looking on these monitors, when he feels to intervene the trainees, how and for what purpose he intervene, what the monitors should include, how it is designed, what are the most important indicators on these monitors, how he is satisfied with design of these monitors etc.

This way of gathering needs was applied for all the tasks with various users/instructors.

The team work on the Simulation Management System, analysed user needs on the Scenario Management System, train, locomotive and railway attributes which the trainee perform on them, monitors in control and watching station and debriefing systems. Same techniques were used by the each team of the project. Each team share their outcomes and comment on them, at the needs definition process.

Simulation Management System is related with all the modules. For example, an instruction is given from the simulation management system affects the dynamic modelling system. In this way possible conflicts of the needs are analyzed and eliminated.

IV.2.3. Presenting User Needs Analysis Findings

The aim is to show users just one simple structure, but to allow for any amount of detail. To do this, arrangement of what users need under chapter headings that represent the users' top-level goals is essential. Users can then see the whole pattern before diving into the details. There should be three levels of headings to organize the requirements: not to be used many more than that (Alexander and Stevens, 2002).

Regarding to the Alexander and Stevens (2002) suggestion to write better requirements, firstly we arranged the customer listed needs which the customer gave us as a formal document. We confront our notes and the document and we listed a new requirements document which we called CRID (Customer Requirements Identification Document).

After an agreement on the CRID, as mentioned that a good structure helps organize the needs, we framed a structure with chapter headings and then fall into the details. In other words, the needs were arranged in documents with structures that provide a hierarchy of headings with separate sections for needs that are closely related. But the arrangement would make it hard to find or understand all the needs related to the single desired result. To overcome this problem we use the TELELOGIC DOORS, requirement management tool.

The structure of Control and Watching Station Needs:

Control and Watching Station

1. Scenario Editor

1.1. Scenario Forming Needs

1.2. Scenario Running Needs

- 1.3. Multifunction Page Needs
 - 1.3.1. General Multifunctions
 - 1.3.2. Valve Multifunctions
 - 1.3.3. Low Voltage Cabinet Multifunctions
 - 1.3.4. Matrix Table
- 1.4. Runfor Monitor Needs
- 2. Monitors on Control and Watching Station
 - 2.1. Monitor 1: Simulator Running Needs
 - 2.2. Monitor 2: Operation Table Needs
 - 2.3. Monitor 3: CGI Road View
 - 2.4. Monitor 4: Trainee Tracking Monitor
 - 2.5. Online Error Detection
- 3. Train, Locomotive and Train Car Needs
 - 3.1. Car Park Page Needs
 - 3.2. Freight Car Page Needs
 - 3.3. Passenger Car Page Needs
 - 3.4. Locomotive Page Needs
 - 3.5. Train Set Page Needs
- 4. Debriefing

If a good structure is created and agreed the user needs should fall into the place without difficulty. After agreeing on the structure we have worked on needs to find out what is really wanted, and to be rewritten so that they are clear and precise. We had written the needs under the headings by using the telelogic doors. We called this new document SRID (System Requirements Identification Document), because

our primary user is only the instructor we did not denote the instructor in each user needs. The examples of needs are listed below;

SRI 258: When forming Scenario, the simulated train could be positioned on railway route in determined conditions.

SRI 259: While forming scenario, maximum 2 vehicles could be positioned on railway route in determined conditions. The vehicle could be varied.)

SRI 315: When forming scenario, the air condition could be determined as rainy, snowy or cloudy.

SRI 316: When forming scenario, rainfall level could be selected.

SRI 317: When forming scenario, snow level could be selected.

SRI 318: When forming scenario, cloudy level could be selected.

SRI 319: When forming scenario, fog level could be selected.

SRI 320: When forming scenario, ground could be selected as dry, wet or covered with snow.

A readable user needs/requirements document does not consist only of needs, no matter how clearly arranged. Requirements are more than pieces of text. Technical documents should have attributes like traceability, need status, related needs, description, risks, verification, validation, source, accuracy, all the way through to completion.

Table IV.2 shows how the needs fall in the structure and some of the attributes. Requirements can be select by using the requirement Id column. If this requirement is also a need, true value is written in the second column. The explanation of the need can be found at the third column. Need State column defines, whether the need is mandatory (M) or optional (O). The effected module is written in the Package in charge column. The next column contains the person, who did the last modification. The last column shows the releated Customer Requirement.

Each System Requirements is related to the Customer Requirements. There can be several System Requirement for a Customer Requirement, because different parts

of the system can be affected from a Customer Requirement. The example of the connection between the Customer Requirements and the System Requirements can be found in the Table IV.2.

Table IV.2: Software Requirements Identification Document

Traceability	Needs ?	ID	Needs Text	Need State M/O	Package in charge	Who modified	CRI
	False	5553	5.1.4.1 SCENARIO EDITOR			Admin	
		6706	The needs of scenario editor are grouped as needs when forming scenario and needs when running scenario. These needs look similar but they are not same in terms of system and functional approach.			Admin	
SRI 253	True	5556	The scenario editor will be represented by two dimensional graphical symbols.	M	Simulation Management System	Admin	CRI 19
	False	5557	5.1.4.1.1 Scenario Forming Needs			Admin	
SRI 254	True	5560	When forming scenario, railway route could be set by selecting beginning and finish parts.	M	Simulation Management System	Admin	CRI 35
SRI 255	True	5561	When forming scenario, the state of rail switches could be set as to straight path or to detour path.	M	Simulation Management System	Admin	CRI 36
SRI 256	True	5562	When forming scenario, the state of 3 lighted small signals could be set as green, yellow, red on yellow, red, flashing green, flashing yellow, flashing red, flashing red on flashing yellow, red on green, or flashing red on flashing green, and the state of 2 lighted small signals could be set as red, yellow, or red on yellow, and the state of 2 lighted small manoeuvre signals could be set as red on green.	M	Simulation Management System	Admin	CRI 37
SRI 257	True	5563	When forming scenario, all signals could be turn on or turn off simultaneously. The mean of turning on all the signals is green lighted and turning off is red lighted.	M	Simulation Management System	Admin	CRI 37
SRI 258	True	5564	When forming scenario, the simulated train could be positioned on any point on the railway road in determined	M	Simulation Management System	Admin	CRI 34

Traceability	Needs ?	ID	Needs Text	Need State M/O	Package in charge	Who modified	CRI
			conditions.				
SRI 259	True	5565	While forming scenario, maximum 2 vehicles could be positioned on railway route in determined conditions. The vehicle	M	Simulation Management System	Admin	CRI 34
SRI 315	True	5621	When forming scenario, the air condition could be determined as rainy, snowy or cloudy.	M	Simulation Management System	Admin	CRI 13
SRI 316	True	5622	When forming scenario, rainfall level could be selected.	M	Simulation Management System	Admin	CRI 13
SRI 317	True	5623	When forming scenario, snow level could be selected.	M	Simulation Management System	Admin	CRI 13
SRI 318	True	5624	When forming scenario, cloudy level could be selected.	M	Simulation Management System	Admin	CRI 13
SRI 319	True	5625	When forming scenario, fog level could be selected.	M	Simulation Management System	Admin	CRI 13
SRI 320	True	5626	When forming scenario, ground could be selected as dry, wet or covered with snow.	M	Simulation Management System	Admin	CRI 13

We listed the needs on telelogic doors as shown in Figure IV.2.

'SIBB' current 1.1 (SIBB) in /TRENDSIM/OUTPUT/SIBB (Formal module) - DOORS											
File	Edit	View	Insert	Link	Analysis	Table	Tools	User	kitchen	Help	
View İhtiyaçBelirleme											
İzlenebilirlik	İster mi?	ID	İhtiyaç Metni	İster durumu	Açıklama	Sorumlu Paketler	Degisikligi Yapan	Is yuku	Risk	Silindi mi?	Silme Nedeni
SIS 917	True	7982	Çalışması esnasında takometrenin sesi duyulacaktır.	opsiyonel		Benzetim motoru	nilgun				
SIS 252	True	5547	Sanal Trenlerin yaklaşma ve uzaklaşma sesleri duyulacaktır.	zorunlu		Benzetim motoru	Administrator	yüks	or	ta	
	False	4699	5.1.4 KONTROL İSTASYONU			Benzetim yönetim sistemi	Administrator				
	False	4700	Kontrol İstasyonlarının sağladığı işlevler birbirini tamamlayıcı nitelikte olduğundan bu bölümde birlikte açıklanmıştır. Bölüm 4.3.4 ve Bölüm 4.3.5'te anlatılanlar ışığında bu iki grubun istemleri bölüm numaralarıyla birlikte aşağıda verildiği gibi istelenmektedir.			Benzetim yönetim sistemi	Administrator				
	False	5553	5.1.4.1 SENARYO EDITÖRÜ				Administrator				
		6706	Senaryo editörüne ait istemler, tanımlama esnasındaki istemler ve kullanımı esnasındaki istemler olarak gruplanmıştır. Bu istemlerin tanımlarında benzerlik görülebilir ancak, fonksiyonel olarak, sistem açısından farklı olduklarından iki bölümde de ayrı ele alınmıştır.				Administrator				
SIS 253	True	5556	Senaryo editörünün grafiksel gösterimi, 2 boyutlu semboller ile olacaktır.	zorunlu			Administrator				
	False	5557	5.1.4.1.1 Senaryo Tanımlama İstemleri				Administrator				
SIS 254	True	5560	Senaryo hazırlama esnasında; başlangıç ve bitiş bölümleri seçilmek suretiyle güzergah belirlenebilecektir.	zorunlu		Benzetim yönetim sistemi	Administrator				
SIS 255	True	5561	Senaryo hazırlama esnasında; seçilen güzergah üzerindeki makasların durumları doğru yola tanzimli veya sapmalı geçişe tanzimli olarak belirlenebilecektir.	zorunlu		Benzetim yönetim sistemi	Administrator				
SIS 256	True	5562	Senaryo hazırlama esnasında; 3 lambalı düce sinyallerin durumları yeşil, sarı, sarı üzeri kırmızı, kırmızı, yanar söner yeşil, yanar söner sarı, yanar söner kırmızı, yanar söner sarı üzeri yanar söner kırmızı, yeşil üzeri kırmızı, yanar söner yeşil üzeri yanar söner kırmızı olarak, 2 lambalı düce sinyallerin durumları kırmızı, sarı veya sarı üzeri kırmızı olarak, iki lambalı düce manevra sinyalinin de durumu yeşil üzeri kırmızı olarak değiştirilebilecektir.	zorunlu		Benzetim yönetim sistemi	Administrator				
SIS 257	True	5563	Senaryo hazırlama esnasında; seçilen güzergah üzerinde istenir tüm sinyaller aynı anda açılıp kapatılabilecektir. Tüm sinyallerin açılması yeşil olarak belirlenmesi, kapatılması ise kırmızı olarak belirlenmesi anlamına gelmektedir.	zorunlu		Benzetim yönetim sistemi	Administrator				
SIS 258	True	5564	Senaryo hazırlama esnasında; gerçek tren seçilen güzergah üzerinde belirlenen koşullar dahilinde istenilen yol konumuna konuşturılabilir.	zorunlu		Benzetim yönetim sistemi	Administrator				
SIS 259	True	5565	Senaryo hazırlama esnasında; seçilen güzergah üzerinde belirlenen koşullar dahilinde istenilen yol konumuna en fazla 2	zorunlu		Benzetim yönetim sistemi	Administrator				

Figure IV.2: User Need Document in Doors

Briefly, we firstly documented the customer written needs CRID. Secondly, we developed detailed need document, called SRID, from original user needs (CRID) and our notes that were taken during the interviews. Thirdly, we went on further detailed needs, called Software Requirements Identification Document, for design and software team that is nothing to do with users. During these step the most important things except the requirements' accuracy, was requirements traceability and change. The requirements management tool Doors help for this issue. Each Software Requirements was linked to related SRID needs and each SRID need was linked to related CRID needs. Changes on needs quite often and they need to be updated. The changes were tracked by using doors as well.

As it is known that use-case diagrams provide a high-level visual representation of the user needs, we also represent the user needs drawing use-case diagrams. The main purpose of a use case diagram is to show what system functions are performed for which actor. In this study we focus on the user needs analysis of control and watching station and scenario management. The main users of the trainer/instructor systems are instructors, instructor represent the actor in our use-case diagrams.

Firstly use-case diagrams of scenario editors, monitors of the control and watching station, car park pages, forming train pages, forming train car pages, and forming locomotive pages were drawn. The detailed use-case diagrams of use-cases in these general topics are drawn. Figure IV.3 depicts what the instructor can do on scenario editor in general view. The boundary of the system is scenario editor.

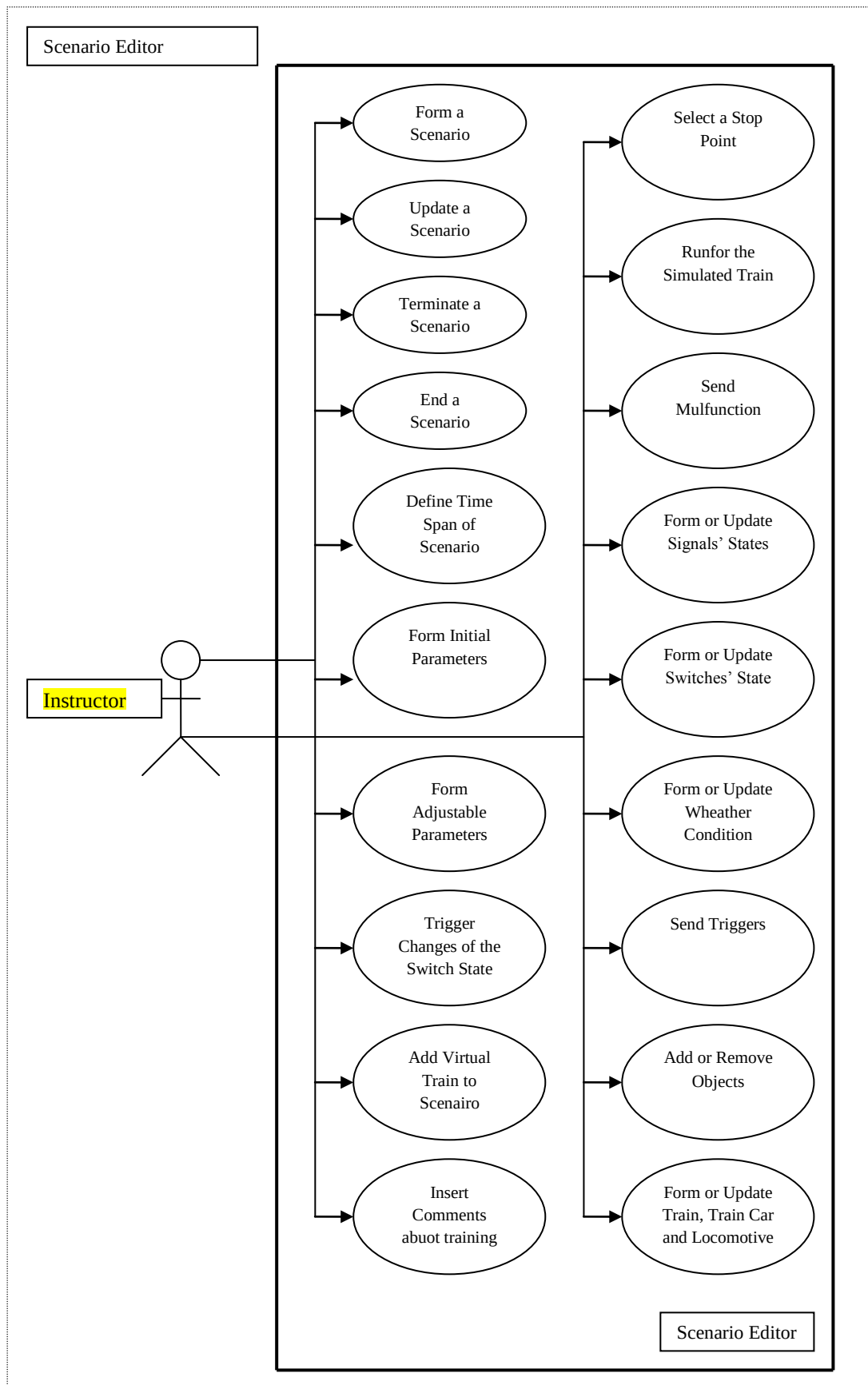


Figure IV.3: Use Case diagram of Scenario Editor

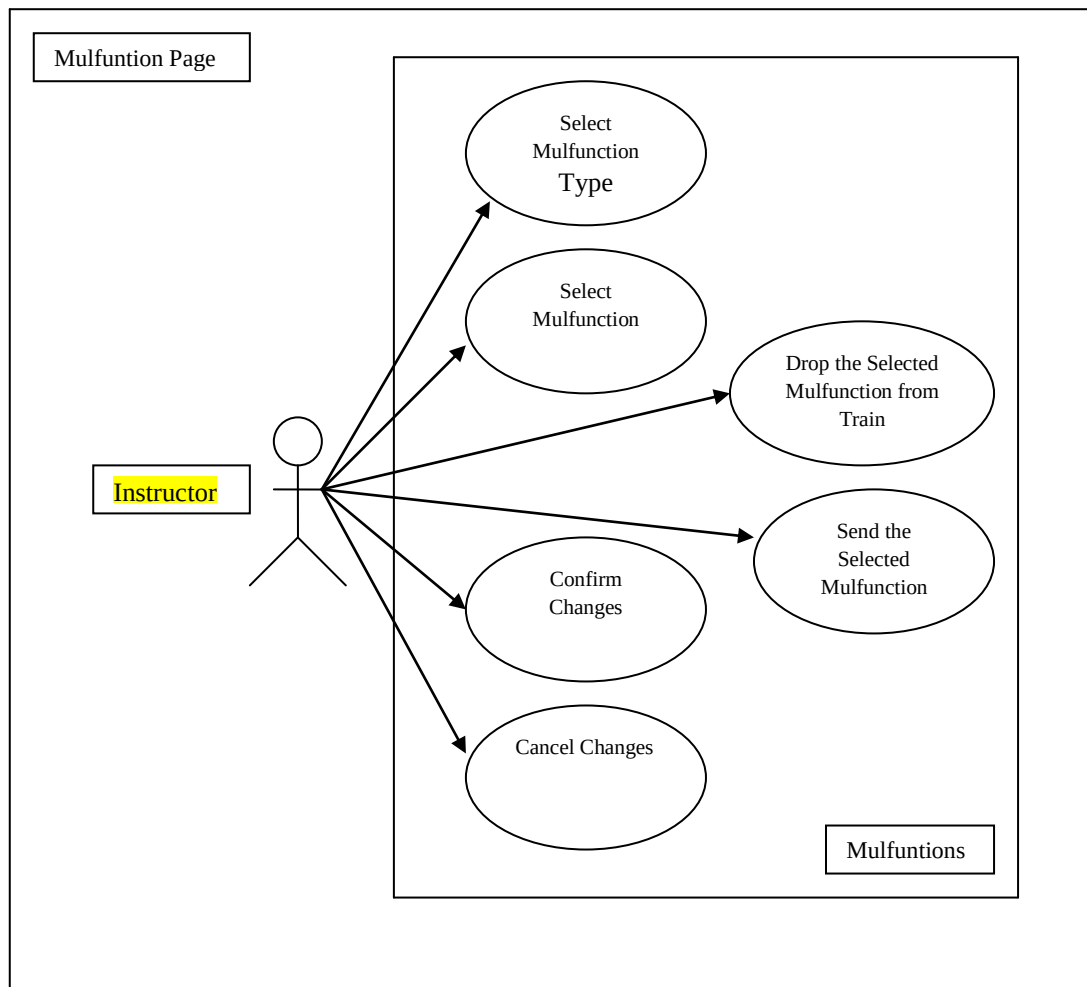


Figure IV.4: Use Case diagram of Multifunction Page

Figure IV.4 depicts what the instructor can do on malfunction page of the system. The boundary of the system is malfunction page. Instructor selects a type of malfunction and a malfunction. Instructor adds the selected malfunction on train or moves the selected malfunction from train. Lastly instructor confirms or cancels these changes.

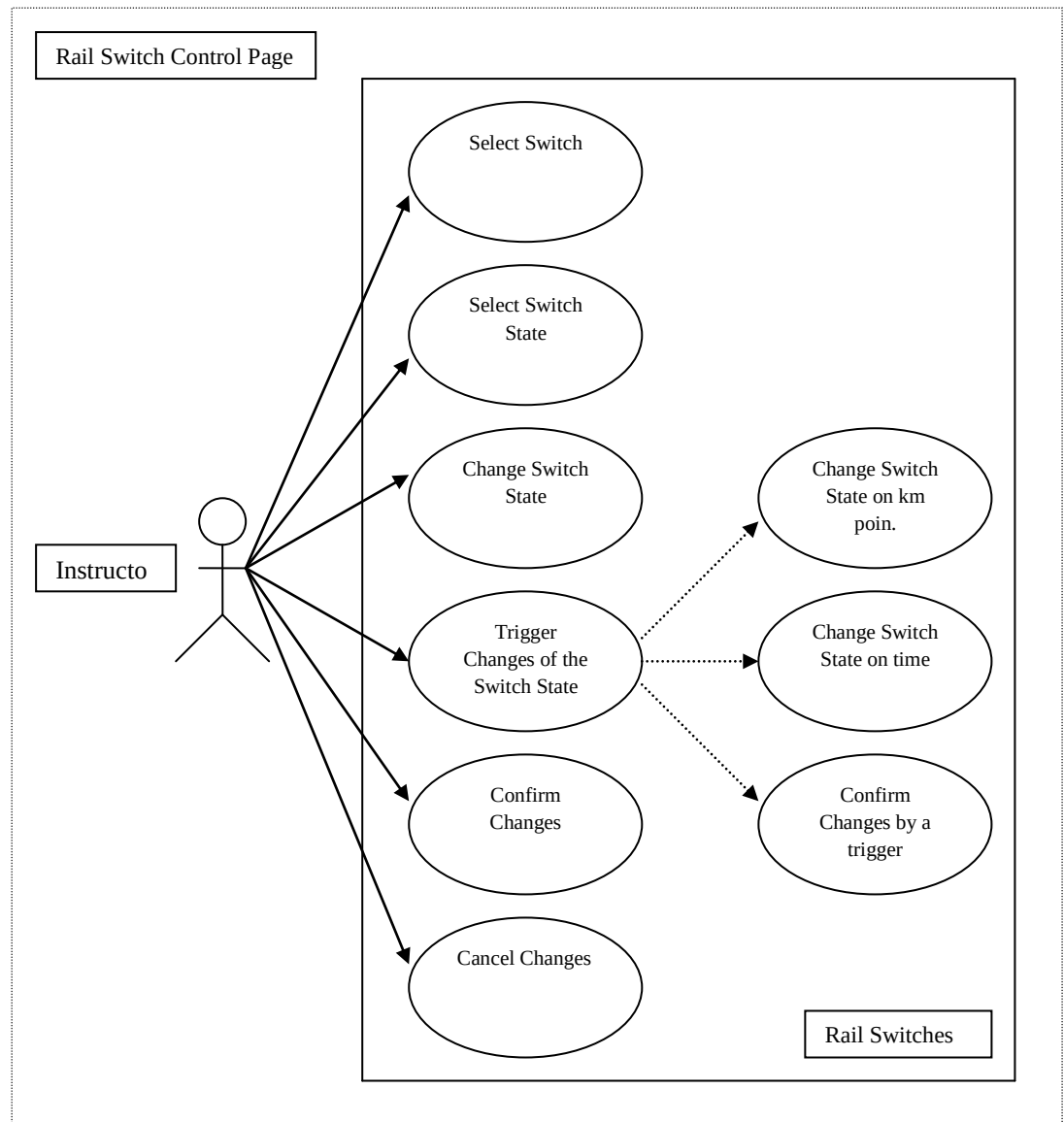


Figure IV.5: Use Case diagram of Rail Switch Control

Figure IV.5 depicts what the instructor can do on rail switch control page of the system. The boundary of the system is rail switch control page. Instructor selects a switch. Instructor selects or changes a switch state. Instructor triggers the selected switch and state according to km point or time or a trigger. Lastly instructor confirms or cancels these changes. Also Figure IV.6 shows the use-case diagram of the rail switch control in Enterprise Architect.

CHAPTER V

CONCLUSION

User Needs Analysis is a multi-disciplinary activity, deploying a variety of techniques and tools at different stages of development and for different kinds of application domain. User Needs Analysis emerges its importance both in User-Centered Design activities which is related to Usability Engineering and Requirements Engineering activities. Both User Centered Design and Requirements Engineering have their own methods and techniques to analyze user needs for software engineering products.

Every project succeeds or fails based on the quality of user needs identified. They set the scope of all subsequent work and tell the project team what the users want. Without good needs, projects fail, are late, come in over budget, or produce systems that are never used. User Need Analysis issues should be fixed early, before committing to a design, because problems caused by poor requirements tend to be deeply embedded in the design and are difficult to remedy afterwards.

The aim of this study is to research the approaches and techniques of user needs analysis and its presenting methods and implementing proper approaches in user needs analysis phase of TRENSIM. In this thesis we will represent the different User Needs Analysis Activities on both UCD and RE methodologies. UCD and RE methodologies should be integrated, or at least aligned for an efficient analysis and the right product needed.

To this end, several user needs analysis activities on both User Centered Design and Requirements Engineering methodologies are represented. Both UCD and RE methodologies focus on users therefore, the first step of user needs analysis process is to identify the stakeholders and the users. The eliciting techniques and presenting way of them are represented on both these methodologies.

As a case study the project called TRENSIM, simulation based training system in rational railway is illustrated. Then the accurate techniques based on specification of the TRENSIM are used to gather and present the user needs. As the aim of the simulation based training is to train machinist in synthetic environment as in real life experiment, the instructors are our main users. Trainees are secondary users who actually do not know what is needed for training purpose on the simulator. The techniques of eliciting user needs used in this project, regarding to its technical aspects and specifications, were interviews, think aloud protocol, contextual inquiry and task analyses. Also the project team was trained as a student to see how the training occurs.

The findings of the user needs analysis are represented in requirements documents and use-case diagrams. The Customer Requirements Identification Document has 100 needs. Scenario Management System has 34 main needs in the CRID document. After customer needs are formed, the needs detailed by the eliciting techniques of think aloud protocol, contextual inquiry and task analyses. The detailed user needs are represented in the System Requirements Identification Document. 881 requirements are listed in the SRID document. 337 System Requirements are formed under the main heading of the Simulation Management System. During the organizing and presenting needs, the most important things except the requirements' accuracy, was requirements traceability and change. The requirements management tool Doors is used for this issue. Changes on the needs were tracked by using doors as well.

The Waterfall approach is followed in the project. According to that approach, after the User Needs Analysis process, the design process is started with defined requirements. However, at any point of the design process, it could be returned to the user needs analysis to change some requirements, so it should not be thought that User Needs Analysis is finished at first step of the project.

In conclusion, there is not a particular way to analyse user needs and represent its findings. Which of these techniques should be used depends on specification of the project, product, and cultural environment of the work. Existing methodologies and techniques appear to be helpful in some but not all aspects of User Needs

Analysis process. Further studies on user needs analysis process should be executed as every project has own speciation, the needs on how proceed and combine techniques of the user needs analysis differ. Appropriate techniques are used at the TRENSIM project and some cases of the project are illustrated in this thesis. The user needs analyses process of the project is completed, but it might be still reviewed at the following stages on the TRENSIM project.

REFERENCES

- Abras, C.; Maloney-Krichmar, D.; Preece, J.: “User-Centered Design” In Brainbridge, W. *Encyclopedia of Human-Computer Interaction*, Sage Puplication, (2004).
- Alexander, I.; Maiden, N.: “Scenarios, Stories, Use Cases: Through The Systems Development Life-Cycle” John Wiley and Sons, (2004).
- Alexander, I.; Stevens, R.: “Writing Better Requirements” Addison-Wesley, (2002).
- Almirall-Hill, M.; Garreta-Domingo, M.: “A user-centered design process to ensure the quality of the new e-classroom”, (2007).
<http://www.eife-l.org/publications/proceedings/ilf07/Contribution182.doc.pdf> (20.05.2009).
- Austin, J.; Delaney, P.F.: “Protocol analysis as a tool for behavior analysis”, *Analysis of Verbal Behavior*, (1998), 15, 41-56
- Barker, R.; Angelopulo, G.: “Integrated Organisational Communication”, Juta and Company Limited, Cape Town, (2005).
- Beyer, H.; Holtzblatt, K.: “Contextual Design: Defining Customer-Centered Systems”, Morgan Kaufmann, San Francisco, CA, (1998).
- Beyer, H.; Holtzblatt, K.; Baker, L.: “An Agile Customer-Centered Method: Rapid Contextual Design”, Springer Berlin / Heidelberg, Book Series Lecture Notes in Computer Science, (2004).
- Bloodworth, A.; Crew, I.: “User-centered design for IT services: The usability toolbox, part1”, University of Berkeley, iNews: Planning, architerture, development, (2007).
<http://inews.berkeley.edu/bcc/Spring2007/998.html> (24.05.2009).

Booth, P. A.: “An introduction to human-computer interaction”, Psychology Press, Erlbaum, **(1989)**.

Brown, K.W.; Cozby, P.C; Kee, D.W.: “Research Methods in Human Development”, Mayfield Publishing Company, London, Toronto, **(1998)**.

Burg, J. F. M.: “Linguistic Instruments in Requirements Engineering.”, IOS Press, Amsterdam, **(1997)**.

Carroll, J.M.: “Scenario-based design: Envisioning work and technology in system development”, JohnWiley & sons, London, **(1995)**.

Chitnis, M.; Tiwari, P.; Ananthamurthy, L.: “Creating Use Case Diagrams”, May 11, **(2009)**.

http://www.developer.com/design/article.php/10925_2109801_1 **(12.05.2009)**

Cockburn, A.: “Writing Effective Use Cases”, Addison-Wesley, Boston, MA, **(2001)**.

Cooper, A.: “The Inmates Are Running the Asylum” Indianapolis, Indiana: SAMS, Division of MacMillan Computer Publishing, **(1999)**.

Constantine, L.L. & Lockwood, L.A.D.: “Software for use: A practical guide to the models and methods of usage-centred design”, Addison-Wesley, Reading, MA, **(1999)**.

Constantine, L.L. & Lockwood, L. A. D.: “Structure and style in use cases for user interface design”, *Object-modeling and user interface design*, in M. van Harmelen (ed), Addison Wesley, Reading, MA, **(2001)**.

Crutcher, R. J.: “Telling what we know: The use of verbal report methodologies in psychological research“, *Psychological Science*, **(1994)**, 5, 241-244.

Deek, G.: “Development of Digital Arabic Content: Incubation Requirements and Training Needs”, *United Nations Economic and Social Commission for Western Asia Information and Communication Technology Division*, **(April, 2008)**.

<http://www.escwa.un.org/divisions/projects/dac/03DAC.pdf> (27.05.2009)

Endsley, M.R.; Bolte, B.; Jones, D. G.: “Designing for situation awareness: an approach to user-centered design”, Taylor & Francis, **(2003)**.

Ericson: “Protocol analysis and Verbal Reports on Thinking”, *An updated and extracted version from Ericsson*, **(2002)**.

www.psy.fsu.edu/faculty/ericsson/ericsson.proto.thnk.html **(24.05.2009)**

Ericsson, K. A.; Simon, H. A.: “Protocol analysis: Verbal reports as data”, MIT Press, Cambridge, MA., **(1993)**.

Ericsson, T.: “Notes on design practice: Stories and prototypes as catalysts for communication”, in J. Carroll (ed) *Envisioning technology: The scenario as a framework for the system development life cycle*, Addison Wesley, Reading, MA., **(1995)**.

Ewusi-Mensah, K.: ‘Critical issues in abandoned information systems development projects’, *Communications of the ACM*, **(1997)**, 40 (9), 74-80.

Feucht, C.; Lamp, J.: “Balance Stakeholder Communication Guide”, **(08.12.2006)**.
<http://balance-eu.org/xpdf/balance-interim-report-no-8.pdf> **(23.05.2009)**

Fowler, M.; Scott, K.: “UML Distilled: A Brief Guide to the Standard Object Modeling Language”, 2nd Edition.; Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, **(2000)**.

Galitz, O. W.: “The Essential Guide to User Interface Design”, John Wiley Publishing, Inc., Third Edition, **(2002)**, 71.

Giacoppo, A. S.: “Developments Methods: User Needs Assessment & Task Analyses”, Department of Psycholgy Catholic University, Washington, **(2001)**.

Goguen, J.; Linde, C.: “Techniques for Requirements Elicitation” *1st IEEE International Symposium on Requirements Engineering (RE'93)*, San Diego, USA, 4-6th January **(1993)**, 152-164

Goguen, J.; Jirotko, M.: “Requirements Engineering: Social and Technical Issues”, Academic Pres, London, **(1994)**.

Gulliksen, J.; Lantz, A.; Boivie, I.: “User Centered Design – Problems and Possibilities”, A summary of the 1998 PDC & CSCW workshop, **(1998)**.
<http://www.sigchi.org/bulletin/1999.2/gulliksen.pdf> **(28.05.09)**

Hackos, J.T. & Redish, J.C.: “User and task analysis for interface design”, John Wiley & Sons, Inc. London, **(1998)**.

Henry, S. L.: “Just Ask: Integrating Accesibility Throughout Design”, Madison, WI: ET/Lawton, **(2007)**.
www.uiAccess.com/JustAsk/

Holtzblatt, K.; Beyer, H., R.: “Requirements Gathering: The Human Factor” *Communications of the ACM*, **(1995)**, 38(5): 31-32.

Holtzblatt, K.; Jones, S.: “Contextual Inquiry”, *Tutorial notes, CHI’92*, Monterey, CA., **(1992)**.

Holtzblatt, K.; “Contextual Design: Experience in Real Life”, *1. Fachübergreifende Konferenz. Stuttgart: B.G. Teubner*, **(2001)**, p.19-22
<http://mc.informatik.uni-hamburg.de/konferenzbaende/mc2001/Holtzblatt.pdf>

Jacobson, I.: “The object advantage: Business process reengineering with object technology”, Addison-Wesley, Reading, MA, **(1995)**.

Jacko, J. A.; Stephanidis, C.; Harris, D.: “Human-Computer Interaction: Theory and Practice”, Lawrence Erlbaum Associates, **(2003)**.

Jacobson, I.; Booch, G.; Rumbaugh, J.: “The Unified Software Development Process” Reading, MA, Addison-Wesley, **(1999)**.

Jacobson, I.; Christenson, M.; Jönsson, P.; Overgaard, G.: “Object-oriented software engineering: A use-case driven approach”, Addison Wesley, Reading, MA., **(1992)**.

Kamandi, A.; Habibi, J.: “Modeling Languages Study and Evaluation Techniques”, *Second Asia International Conference on Modelling & Simulation*, 13-15 May, **(2008)**, 553, 2

Kirakowski, J.: “Questionnaires in Usability Engineering: A List Frequently Asked Questions”, 3rd Edition, (2nd June, **2000**).
<http://www.ucc.ie/hfrg/resources/qfaq1.html> **(27.05.2009)**

Kotonya, G.; Sommerville, I.: “ An introduction to requirements engineering”, 05 May, **(1998)**.
http://www.computing.dcu.ie/~nbrophy/ca222/week4_6/som1.pdf **(14.05.2009)**

Kuhl, F.; Weatherly, R.; Dahmann J.: “Creating computer simulation systems”, Prentice Hall, New Jersey, **(2000)**.

Laplante, P.A.: “What Every Engineer Should Know about Software Engineering”, CRS Press I Llc, **(2007)**, 44.

Lehman, M.M.: “Programs, Life Cycles, and Laws of Software Evolution”, *Proceedings of the IEEE*, **(1980)**, 68(9): 1060-1076.

Leffingwell, D.; Widrig, D.: “Managing Software Requirements: A Unified Approach”, Addison Wesley, **(1999)**.

Leffingwell, D.; Widrig, D.: “Managing Software Requirements: A Use Case Approach”, Addison Wesley, **(2003)**.

Lewis, C.H.: “Using the “think aloud” Method in Cognitive Interface Design”, IBM Research Report, RC-9265, Yorktown Heights, New York **(1982)**.

Lewis, C.; Rieman, J.: “Task-Centered User Interface Design: A Practical Introduction”, **(1994)**

Lindgaard, G.: “Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems”, Chapman and Hall, London, **(1994)**

Lindgaard, G.; Dillon, R.; Trbovich, P.; White, R.; Fernandes, G.; Lundahl, S.; Pinnamaneni, A.: “ User Needs Analysis and requirements engineering: Theory and practice”, *Science Direct*, **(2006)**, 47-70

Lubars, M.; Potts, C.; Richter, C.: “A review of the state of the practice in requirements modelling”, *Proceedings 1st International Symposium on Requirements Engineering*, in S. Fickas & A.C.W. Finkelstein (eds), IEEE Computer Society Press, Los Alamitos, CA., **(1993)**.

Luczak, M.: “High-tech training tools, Railway Age”, **(2000)**.
http://www.findarticles.com/p/articles/mi_m1215/is_7_201/ai_64337987

Maamar, Z.: “Design of a simulation environment based on software agents and the high level architecture”, *Information and Software Technology, Elsevier*, **(2003)**. 45. 137-148

Maguire, M.: “RESPECT user-Requirements framework handbook”, HUSAT Research Institute, Loughborough University, UK, **(1997)**.

Maguire, M.: “Methods to support human-centered design”, HUSAT Research Institute, Loughborough University, UK, **(2001)**.
<http://www.idealibrary.com> **(23.05.09)**

Maunder, A.; Marsden, G.; Gruijters, D.; Blake, E.: “Designing Interactive Systems for the Developing World Reflections on User Centered Design”, May 13, **(2007)**.
<http://people.cs.uct.ac.za/~edwin/MyBib/2007-maunder.pdf>

Mauro, L. C.: “User-Centered Design methodology”, May 10, **(2009)**.
http://www.taskz.com/ucd_indepth.php **(15.05.09)**

Mayhew, D.J.: “The usability engineering lifecycle: A practitioner’s handbook for user interface design”, Morgan Kaufmann, Inc. San Francisco, CA, **(1999)**.

Mayhew, D.J.: “Requirements specifications within the usability engineering life cycle”, in J.A. Jacko & A. Sears (eds) *The human-computer interaction handbook*:

Fundamentals, evolving technologies and emerging applications, Lawrence Erlbaum Associates, Mahwah, NJ, **(2003)**.

Mazza, C.; Fairclough, J.; Melton, B.; De Pablo, D.; Scheffer, A. ; Stevens, R.: ” Software engineering standards” Prentice-Hall, New York, **(1994)**.

Mephram, S., “Synthetic Environments - Delivering Real Benefits for UK Defence”, *Simtect 98 Proceedings*, **(1998)**.

Merriam-Webster’s Online Dictionary: “Questionnaire”, **(2008)**.
<http://www.merriam-webster.com/dictionary/questionnaire> **(22.05.09)**

Nemeth, C. P.: “Human factors methods for design: making systems human-centered”, CRC Press, **(2004)**.

Neuman, W.L.: “Social Research Methods: Qualitative and Quantitative Approaches”, Allyn and Bacon, Boston, **(2000)**.

Nielsen, J.: “ Field Studies Done Right: Fast and Observationl”, Adapted from Jakob Nielsen’s Alertbox, **(2002)**.
<http://www.useit.com/alertbox/20020120.html> **(23.05.09)**

Norman, D.A.: “The Psychology of Everyday Things”, Basic Books, New York, **(1988)**.

Normon, D.A.: “The Design of Everyday Things”, Doubleday, New York, **(1998)**.

Norman, D.A.; Draper, S.W.: “User-Centered System Design: New Perspective on Human-Computer Interaction”, Lawrence Earlbaum Associates, Hillsdale, **(1986)**.

Nusibeh, B.; Easterbrook, S.: “Requirements Engineering: A Roadmap”, In A. C. W. Finkelstein, A. (ed), *The Future of Software Engineering, companion volume to the proceedings of the 22nd International Conference on Software Engineering, ICSE'00*, IEEE Computer Society Press, **(2000)**.
<http://www.cs.toronto.edu/~sme/papers/2000>

Özbek, M.: “Etmen Tabanlı Zeki Öğretim Sistemi Geliştirme”, *Doktora Tezi*, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, Türkiye, **(2007)**.

Posner, M.I.: “Foundations of Cognitive Science” MIT Pres, **(1993)**.

Schuler, D.; Namioka, A.: “Participatory design: Principles and practices”, Lawrence Erlbaum Associates, Mahwah, NJ., **(1993)**.

Si Alhir, S.: “ Understanding Use Case Modeling”, Published in *Methods & Tools, An international software engineering digital newsletter*, Martinig & Associates, **(2000)**.

Simon H. A.; Kaplan C. A.: “Foundations of cognitive science”, MIT Press, Cambridge, MA., **(1989)**.

Sommerville I.; Sawyer P.: “Requirements Engineering: A Good Practice Guide”, John Wiley and Sons, **(1997)**.

Stoney, G.: “Document Your User Needs”, **(2008)**.

<http://greatengineering.net/System-Engineering/Stakeholder-Requirements/Document-Your-User-Needs.html>

Sudman, S.; Bradburn, N.M.; Schwarz, N.: “Thinking about answers: The application of cognitive processes to survey methodology”, Jossey-Bass, San Francisco, CA, USA, **(1996)**.

<http://www.surveypedia.com>, **(27.05 2009)**

Sutcliffe, A.: “User-centred Requirements Engineering: Theory and practice”, Springer Verlag, London, **(2002)**.

Thomas, C. & Bevan, N.: “Usability context analysis: A practical guide”, NPL Usability Services, National Physical Laboratory, Teddington, UK, **(1996)**.

TRENSIM, “Sanal Ortam Tren Sürüş Eğitimi için Milli Benzetim Sistemlerinin Geliştirilmesi Projesi Sözleşme Dokümanı” TÜBİTAK Proje No: 106G137, Classified Document, **(2007)**.

TRENSIM, “Sanal Ortam Tren Sürüş Eğitimi için Milli Benzetim Sistemlerinin Geliştirilmesi Projesi Öğrenci ve Öğretmen İsterleri Analiz Raporu” TÜBİTAK Proje No: 106G137, Classified Document, **(2008)**.

USAID Center for Development Information and Evaluation: “Conducting Focus Group Interviews: Performance Monitoring and Evaluation”, **(1996)**.
http://pdf.usaid.gov/pdf_docs/PNABY233.pdf, **(25.05 2009)**

van Lamsweerde, A.; Darimont, R.; Letier, E.: “Managing conflicts in goal-driven requirements engineering” IEEE Transactions on Software Engineering, **(1998)**,24(11): 908-926.

Vredenburg, C.; Isensee, S.; Righi, C.: “ User-centered design: An integrated approach”, Prentice-Hall, Upper Saddle River, NJ.**(2002)**.

Wieggers, Karl E.: “Software Requirements 2: Practical techniques for gathering and manging requirements through the product development cycle”, Redmond, Microsoft Press, **(2003)**.
<http://www.processimpact.com>. **(25.05.09)**

Zave, P.: “Classification of Research Efforts in Requirements Engineering” *ACM Computing Surveys*, **(1997)**, 29(4): 315-321.

CURRICULUM VITAE

Dilek AĞIR was born in Istanbul in 1979. After completing his primary education in Istanbul at Gazi Primary School, she attended Çapa Secondary School. Then she was graduated from Şehremi High School.

In year 1998, she was admitted to Industrial Engineering Department of Sakarya University. She was graduated from Sakarya University with first degree. She had attended English language classes in Esher Adult Education Center for one year and had attended Academic English language classes in South Thames College in London for one year.

In year 2006, she was admitted to postgraduate degree of Industrial Engineering Department of Marmara University. Also she had been employed as a researcher of TRENSIM project which has been conducted by TUBITAK and MARMARA UNIVERSITY. She had worked as a system analyst and Designer in the TRENSIM for 18 months. During the project she had studied on User Needs Analysis and prepared her thesis on this subject.