A SOFTWARE FUNCTIONALITY MODEL FOR FUNCTIONAL SIZE MEASUREMENT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BARIŞ ÖZKAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
PHILOSOPHY OF DOCTORATE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

SEPTEMBER 2012

A SOFTWARE FUNCTIONALITY MODEL FOR FUNCTIONAL SIZE MEASUREMENT

Submitted by **BARIŞ ÖZKAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Information Systems, Middle East Technical University** by,

Prof. Dr. Nazife Baykal ________________________________
Director, **Informatics Institute**

Prof. Dr. Yasemin Yardımcı Çetin ________________________________
Head of Department, **Information Systems**

Prof. Dr. Onur Demirörs ________________________________
Supervisor, **IS, METU**

Examining Committee Members:

Prof. Dr. Semih Bilgen ________________________________
EEE, METU

Prof. Dr. Onur Demirörs ________________________________
IS, METU

Assoc. Prof. Dr. Ali Doğru ________________________________
CENG, METU

Assoc. Prof. Dr. Altan Koçyiğit ________________________________
IS, METU

Assist. Prof. Dr. Özgür Tanrıöver ________________________________
C.Eng., Ankara University

Date: 12 September 2012

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

Name, Last Name: BARIŞ ÖZKAN

Signature :

ABSTRACT

A SOFTWARE FUNCTIONALITY MODEL FOR FUNCTIONAL SIZE MEASUREMENT

Özkan, Barış
Ph.D., Department of Information Systems
Supervisor : Prof.Dr. Onur Demirörs

September 2012, 256 pages

Functional size is among the few software size measures for which well-structured and standardized methods exists for its measurement. Although Functional Size Measurement(FSM) methods have gone a long way, one ongoing criticism on FSM methods is the discrepancies in the measurement results of the same software obtained by different measures. In this thesis study the sources of discrepancies that involve the functional properties of measurands and constructs of the FSM method models are investigated in two exploratory case studies. In the light of the findings, a software functionality model for functional size measurement is proposed. The model is founded on a characterization of software functionality from a requirements engineering point of view and it aims is to facilitate reliable size measurements on the basis of formalized concepts and rules. Two case studies are conducted in order to evaluate the applicability of the model and validate its effectiveness.

Keywords: Software size measurement, Functional size measurement, Software functionality

ÖZ

İŞLEVSEL BÜYÜKLÜK ÖLÇÜMÜ İÇİN BİR YAZILIM İŞLEVSELLİK MODELİ

Özkan, Barış
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi : Prof.Dr.Onur Demirörs

Eylül 2012, 256 sayfa

Yazılım işlevsel büyüklüğü, ölçümü için iyi yapılandırılmış ve standartlaştırılmş yöntemler var olan nadir büyüklük ölçülerindendir. İşlevsel büyüklük ölçme yöntemleri hayli yol almışsa da, ölçümcülerin aynı yöntemi izleyerek aynı yazılım için elde ettikleri ölçümlerdeki farklılıklar hala bir eleştiri konusudur.

Bu tez çalışmasında bu farklılıkların yöntemlerde kullanılan modeller ve ölçülen gereksimler ile ilgili sebepleri iki durum çalışması ile incelenmiştir. Bulgular ışığında, işlevsel büyüklük ölçümü için bir yazılım işlevsellik modeli önerilmiştir. Model gereksinim mühendisliği bakışından tanımlı bir işlevsellik üzerine kurulmuştur ve modelin amacı formalleştirilmiş kural ve kavramlar üzerinden güvenilir ölçümler yapılmasının sağlanmasıdır. Modelin uygulanabilirliğinin ve etkinliğinin geçerlenmesi için iki durum çalışması gerçekleştirilmiştir.

Anahtar Kelimeler: Yazılım büyüklük ölçme,İşlevsel büyüklük ölçme, Yazılım işlevseliği

To my dear parents and to my lovely wife

ACKNOWLEDGEMENTS

It is difficult to overstate my gratitude to my Ph.D. supervisor, Onur Demirörs. His great inspiration, enthusiasm and his efforts to explain things simply and clearly stimulated the ideas in this thesis. I would have been lost without his encouragement, generous advice and good company.

I want to express my thanks to my committee members Semih Bilgen and Ali Doğru for their valuable suggestions and comments.

My sincere thanks to Oktay Türetken, Özden Özcan Top , Erdir Ungan and many other friends at Software Management Research Group. This research would not have been possible without their collaboration and support.

Many thanks to students at Informatics Institute for participating in the case studies as part of this research.

Finally, I am deeply grateful to my parents and my sister for their endless support and encouragement during these years. They were always with me whenever I needed them. My dearest thanks go to my wife, Bilge. She shared the burden of my worries and concerns in all stages of my PhD study. She contributed in reviewing and proofreading the thesis and offered me valuable suggestions. I am eternally grateful for her unwavering support, patience and love. Without her encouragement and understanding, I am not sure I could have done it.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS / ABBREVIATIONS

1NF	:First Normal Form
2NF	:Second Normal Form
3-D	:Three-Dimensional
3NF	:Third Normal Form
4FSM	:Software Functionality Model for Functional Size Measurement
API	:Application Programming Interface
AR	:Accuracy Rate
ASSET- R	:Analytical Software Size Estimation Technique Real-Time
B	:B Specification language
C	:Capture
BFC	:Base Functional Component
CCS	:Calculus of Communicating systems
CDR	:Component Detection Rate
CFP	:COSMIC Function Point
CMMI	:Capability Maturity Model Integrated
COCOMO	:Constructive Cost Model
COSMIC	:Common Software Measurement International Consortium
CPM	:Counting Practices Manual
CRM	:Customer Relationship Management
CSP	:Communicating Sequential Processes
DET	:Data Element Type
DFD	:Data Flow Diagram
E	:Entry
E&Q	:Early and Quick
EFP	:Early Function Points
EI	:External Input
EIF	:External Interface File
EO	:External Output
EQ	:External Query
E-R	:Entity - Relationship
FFP	:Full Function Points
FiSMA	:The Finnish Software Metrics Association
FM	:Functional Message
FMC	:FunctonalMessageCapability
FP	:Function Point
FPA	:Function Point Analysis
FSM	:Functional Size Measurement
FUR	:Functional User Requirement
GUI	:Graphical User Interface
IEC	:International Electrotechnical Commission
IFPUG	:International Function Point Users Group
ILF	:Internal Logical File
IS	:Infomation System

ISBSG	:International Software Benchmarking Standards Group
ISO	:International Organization for Standardization
LOC	:Lines of Code
LT	:Logical Transaction
MK II FPA	:Mark II Function Point Analysis
MM	:Movie Manager Application
NEFPUG	:The Netherlands Function Point Users Group
NESMA	:The Netherlands Software Metrics Users Association
OO	:Object Oriented
OOFP	:Object Oriented Function Points
OOI	:Object of Interest
OOmFP	:Object Oriented Method Function Points
OP	:Object Points
POPs	:Predictive Object Points
R	:Read
RDBMS	:Relational Database Management System
Rel.Std.Dev	:Relative Standard Deviation
RET	:Record Element Type
ROOM	:RealTime Object Oriented Model
RQ	:Research Question
RRRT	:Rational Rose Real Time
SDM	:Surrogate Data Model
SMRG	:Software Management Research Group
SPR	:Software Productivity Research
S-R	:Stimulus-Response
SRS	:Software Requirements Specification
SSM	:Software Sizing Model
Std.Dev.	:Standard Deviation
TF	:Transactional Function
U	:Update
UML	:Unified Modeling Language
VDM-ML	:Vienna Development Method-Meta Language
W	:Write
WOS	:Web Ordering Software Application
X	:Exit
XML	:Extended Markup Language
Z	:Z Specification Language

LIST OF SYMBOLS

$=$	:equals
$\twoheadleftarrow\!\twoheadrightarrow$	:total surjective relation
$\leftarrow\!\twoheadrightarrow$	:partial surjective relation
$A \rightarrow D$	:D is functionally dependent to A
$\mathbb{I}$	:the set of integers
$\in$	:is a member of
$\notin$	:is not a member of
X	:cartesian product
$\{a\}$	:the singleton set of a
$\{a, b, c\}$	:the set with members set of a, b, c
$\cup$	:set union
$\cap$	:set intersection
$\setminus$	:set minus
$\emptyset$	:empty set
$\mathcal{P}_{\geq t}(A)$	:the set of subsets of A of cardinality greater than t
$\langle a, b, c\rangle$	:n-tuple
$a[b]$	:the component b of n-tuple a
$P \wedge Q$	:P and Q
$P \vee Q$	:P or Q
$\forall x$	:for all x
$\exists x$	:there exists an x
$\neg P$	:not P
$\subseteq$	:subset
$\subset$	:proper subset
$(x : A \| P(x))$	:the set of x in A such that $P(x)$
$\circledast$	:set structure constructor
$\otimes$	:tuple structure constructor
$\blacksquare$	:end of example
$\square$	:end of definition

CHAPTER 1

INTRODUCTION

Software size is a widely accepted predictor of software projects and size measurement is vitally important to software management [9] [10]. From the times where software size meant ”lines of code” for the majority of software engineers, the understanding of software size have evolved into a multi-aspect concept in parallel to the developments in software engineering methods and software measurement research. Software size has been refined to be measured in different properties such as functionality delivered to the software users, length of code, complexity and amount of reuse [11] [12] and sizing concerns have been distinguished with respect to user’ s viewpoint and the developer’s viewpoint [1].

The idea of measuring software functionality was first introduced with the Function Point Analysis (FPA) method [13]. Since then, not only the original FPA was adapted but also FPA variants and new functional size measurement (FSM) methods have been developed in order to meet theoretic and practical measurement requirements in emerging contexts [7][3]. Availability of the software artifacts that describes software functionality in the early phases of software development and the independence from development methods and technical implementation decisions made functional size an attractive measure to be used in effort estimation and many other software engineering practices [14] [15] [16] [17] [18] [19].

A desired property of any measurement method is to facilitate measurements such that measurers can measure a specific attribute on a specific entity consistently and repeatably [12] [1]. These characteristics are essential for an agreement on the reliability of a method where the provided measure is independent of the measurer and the environment. However,one significant challenge of FSM methods remains to be the discrepancies in the measurement results and openness to subjective interpretations[20] [21] [22] [23] [24] [25] [26] [27] [28] [29].

Among others, one core factor that allows subjective results is the lack of formalism in the descriptions of the software models the FSM methods rely on.

In this research we explore factors that lead to variations in FSM results and propose a formalized software functionality model for FSM (4FSM) to improve consistency in measurement results on the basis of our findings. We demonstrate how measurements can be performed reliably using 4FSM in a FSM method.

1.1 The Context

Effective software engineering requires measuring the attributes of software products in order to understand,control and predict. Project managers need measures to estimate how much software to build, in what time and at what cost. Process managers need measures to compare before and after productivity figures for a process improvement initiative or to normalize various process metrics based on a selected product attribute. They all need objective, comparable, thus, reliable measures in order to benchmark their projects across organizations. Software size has been such a software measure which has been widely accepted to be a determinant software effort and duration and has been used as a primary input to software estimation and productivity models.

In section 1.1.1, we summarize properties of software size and give insight into the measures that quantify length and functionality attributes of software. Section 1.1.2 explains the steps of a FSM process and highlights their important artifacts. Section 1.1.3 summarizes general properties and concepts of FSM methods.

1.1.1 Software Size

With the emergence of new software engineering methods, techniques and implementation technologies, size have been regarded to have multiple dimensions distinguished from several different aspects. Fenton[11] defines length, functionality, reuse and complexity as dimensions of size such that each captures a key aspect of software size analogous to height, width and volume of a physical entity. Among them length and functionality measures are the most popular and used.

Length. Length is the physical size of the product. Lines of Code (LOC) is the most commonly used and oldest traditional size measure. It has been regarded as an intuitive measure of software product size and relatively easy to capture. LOC can be accurately measured only at the later stages of a project after the code is available or it was estimated by expert methods. LOC has been used in many different ways distinguished by how blank lines, commented lines, data declarations, separate instructions are handled[11]. Another factor of variation is the separation of delivered LOC from the written code. As an alternative to LOC for measuring the code length Halstead proposed a set of metrics that captures several attributes of a program [30]. He defined an algorithm as a collection of tokens that comprises operators or operands. The basic measures for these tokens are: $\mu 1$= the number of distinct operators
$\mu 2$= the number of distinct operands
N_1= the total occurrences of operators
N_2= the total occurrences of operands
For a program P three Halstead metrics are:
$Length(N) = N_1 + N_2$
$Vocabulary(\mu) = \mu 1 + \mu 2$
$Volume(V) = N * \log_2 \mu$
Halstead's "Software science" has been the subject of many criticisms[31][11]. They include:

- Theoretical shortcomings,
- Lack of consensus among researchers on classification and counting operators and operands,
- Counting scheme being language dependent and imprecise while mapping to new generation languages e.g object-oriented concepts,
- Validation with very small sample sizes and programs.

However, Halstead's work was found instrumental in introducing metrics to computer scientists and has been argued to be reasonable such that the attributes vocabulary, length, and volume reflect different views of size from the measurement theory point of view [11].

Although LOC measures have been widely used in various estimation productivity models and process metrics, it has received many criticisms due to its limitations[32][33][34] [1] . They include:

- inadequacy for consistent usage in project,team and individual productivity comparisons and benchmarking
- being measured in many different ways lacking standardization
- language dependency
- late availability and difficulty in estimating LOC
- diminishing utility due to increased number of programming languages and ambiguity in handling coded concepts (e.g, data attributes, classes, methods,inheritance,reuse) by new generation languages

Other length measures include number of bytes used for storing the program text and number of characters in the program text which are not as popular and widely used as LOC measures.

Functionality. In contrast to the length measures which quantify technical and physical software artifacts, functionality measures quantify software as seen from the user's point of view and from the concepts that have a direct meaning to software users. Albrecht was the first to propose the idea of measuring software functionality in his FPA method [13]. Albrecht explains his motivation as follows:

> *"To measure productivity we had to define and measure a product and a cost. The product that was analyzed was function value delivered. The number of inputs, inquiries, outputs, and master files delivered were counted, weighted, summed, and adjusted for complexity for each project. The objective was to develop a relative measure of function value delivered to the use that was independent of the particular technology or approach used" [13, p.84]*

After its introduction the original FPA method has become popular in software engineering community[7] and a group of international interest, International Function Point Users Group (IFPUG) standardized the method. In the following years, variants of FPA and new methods have been proposed to overcome the shortcomings of FPA, to adapt the FPA approach to different development environments, domains and methods and to estimate functional size earlier in project lifecycles. Table 1.1 gives a list of methods from the literature that measures software functionality.

Table 1.1: Methods that Measure the Functionality Attribute of Software

Year	Method Name	ISO Certification
1979	Albrecht/IFPUG FPA[13]	✓
1982	DeMarco's Bang Metrics[35]	
1986	Feature Points[36]	
1988	Mark II FPA[37]	✓
1990	NESMA FPA[38]	✓
1990	ASSET- R[39]	
1992	3-D Function Points[40]	
1994	Object Points [41]	
1994	FP by Matson,Barret and Mellichamp [42]	
1997	Full Function Points[43]	
1997	Early FPA [44]	
1998	Object Oriented Function Points[45]	
1999	Predictive Object Points[46]	
1999	COSMIC Full FP [47]	✓
2000	Web Object Points [48]	
2000	Early&Quick COSMIC FFP[49]	
2000	Kammelar's Component Object Points[50]	
2001	OOmFP [51]	
2004	FiSMA FSM [52]	✓

As a consequence to the increasing number of methods each having a different view on functionality attribute of software the need for a common vocabulary increased. In response, an ISO working group was started to establish common principles, definitions and norms for FSM methods. The endeavor resulted in ISO 14143 six-part standard (Table 1.2). As of today there are five ISO conformant FSM methods published as ISO standards [53][54][55][56][57]. One significant contribution of ISO 14143 has been in the resolution of the ambiguity between functional size as an attribute measure and as an effort measure,thus, underlining the difference between the practice of measuring functional size and using functional size for effort and cost estimation purposes. Today, among various size measures, functional size is regarded as a versatile measure having various uses in many software engineering practices such as estimation and acquisition[17] [18]. Thanks to the well-structured FSM methods, their improvements with respect to measurement theory and the standardization process, it has been taken beyond a pragmatic practice [1]. Functional size is also used the primary size measure in major public benchmark datasets [64].ISO compliant FSM methods have international communities that form bodies for setting the standards where the communities provide a mechanism for improvement of the methods by responding to empirical and theoretical

Table 1.2: ISO/IEC 14143 Standard Set

14143-1:Definition of concepts [58]
14143-2:Conformity evaluation of software size measurement methods to ISO/IEC 14143-1 [59]
14143-3:Verification of functional size measurement methods[60]
14143-4:Reference model[61]
14143-5:Determination of functional domains for use with functional size measurement[62]
14143-6:Guide for use of ISO/IEC 14143 series and related International Standards[63]

feedback [65][66][67].

1.1.2 Measurement Process for FSM

The context for software measurement process can be viewed as an ordering of steps and interrelated activities that are distinguished by their outputs [1] [68].

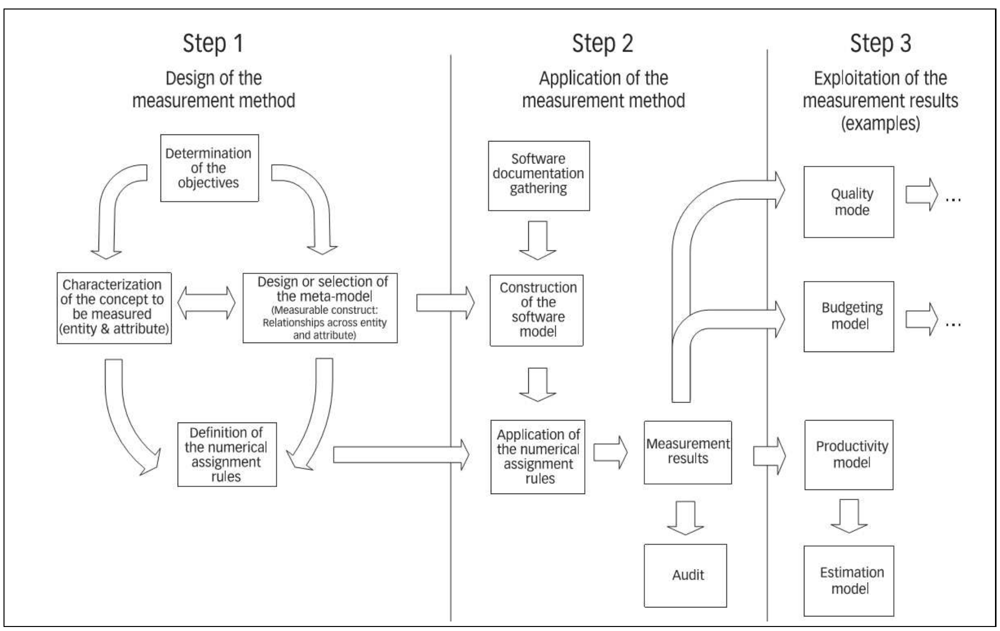


Figure 1.1: Measurement Context Model-adapted from [1]

Step-1. The first step involves the design of a measurement method. In this step either a measurement method that suits the need is selected or a new method is designed. The input of this step is a measurement objective and the output is a generic method that comprises the specification of the measurable concepts and constructs, their relations and the rules for assigning numbers to the constructs. A measurement method characterizes the entity to be measured and its attribute to be measured. Entities are the objects we observe in the real world and attributes are the properties an entity possesses. The characterization can be made by decomposing the concept into sub-concepts and specifying the role of each sub-concept in the constitution of the concept to be measured. The decomposition should describe how these sub-concepts are defined. The method has such a form that the interrelations between sub-concepts are organized into a measurable construct, e.g. a meta-model(Fig 1.2).The method must have descriptions for the meta-model such that they must be generic and it should tell how to identify the constructs. The assignment of numeric values to the concept involves definition of an empirical relation set and the selection of a measurement unit. In ISO 14143 the measurable construct is termed as Base Functional Component (BFC) and a category of BFCs is termed as Base Functional Component Type (BFC Type) [58].

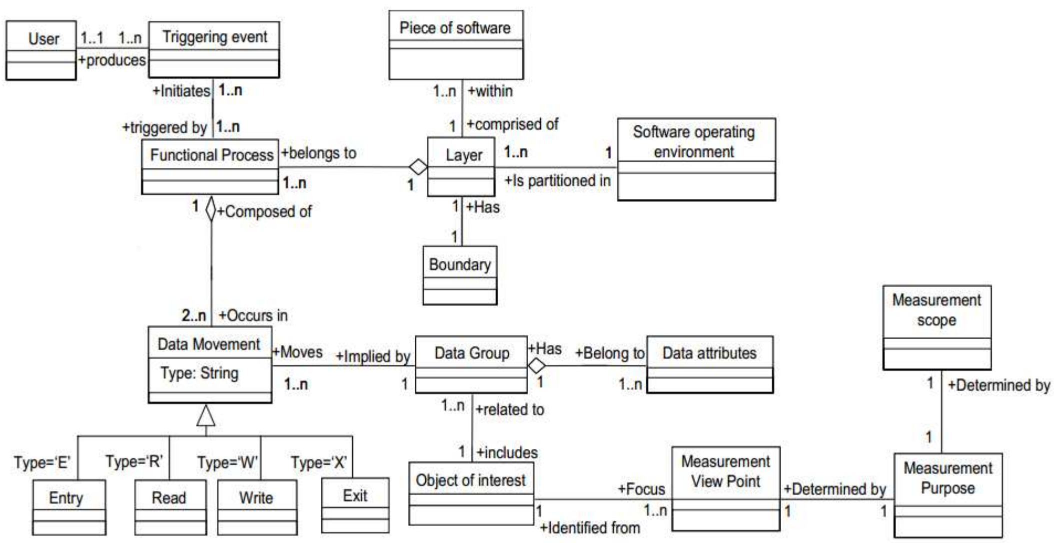


Figure 1.2: COSMIC FSM Meta Model-adapted from [2]

Step-2. The second step involves the application of the method to a measurand (in our case to a piece of software) in a special context of measurement by following the method rules. The output is the specific measurement result obtained. This step involves gathering the software artifacts (e.g documents, diagrams, code) and then constructing a model in accordance with the method meta-model. This is followed by assigning numbers to the constructs following the method numbering scheme. Finally, the measurement results are verified to ascertain their quality and presented in a selected form for use(e.g. documentation, measurement report, measurement record). The primary output is a numeric value assigned to the concept(e.g functionality).

Step 3. In this step the measurement result is exploited. That is, it is put in a quantitative or qualitative model individually or in combination with other simple or derived measures or metrics (e.g. effort, productivity ratios, LOC, team-size, defect count). Although FSM methods refer software ”functionality” as the concept to be measured, each method rely on a different software meta-model that comprises different constructs, associated relations and a number assignment scheme. Thus, each method defines a separate measure for the functionality attribute of software and there does not exist a single universal way of measuring software functional size. ISO 14143 FSM standard does not define any meta-model but sets the principles,norms and criteria for a compliant FSM method.

1.1.3 General Characteristics of FSM Models

The most widely used FSM methods have Albrecht’s FPA as the origin and they share a similar view on the high level concepts that represent software functionality [69] [3]. Figure 1.3 illustrates the similar concepts between IFPUG, COSMIC and MARK II FSM methods and their associations[3] The constituent parts of FSM models can be grouped and studied under the following headings:

Transactional Concepts. Transactions characterize the actions of software that can be controlled by the software users and other entities in the environment. IFPUG defines ”transactional function” as an elementary process that provides functionality to the user to process data where an ”elementary process” is defined as ”the smallest unit of activity that is meaningful to the user and constitutes a complete transaction” [54]. MARK II FPA defines ”logical transaction” as the basic functional component which is the smallest and complete unit of information processing that is meaningful to the end user in the business. It is triggered by an event in the real world of interest to the user, or by a request for information[55]. Similarly COSMIC FSM uses ”functional process” concept to represent software actions that are triggered by functional users for informing the piece of software that the functional user has identified a triggering event in the environment. A functional process is complete when it has

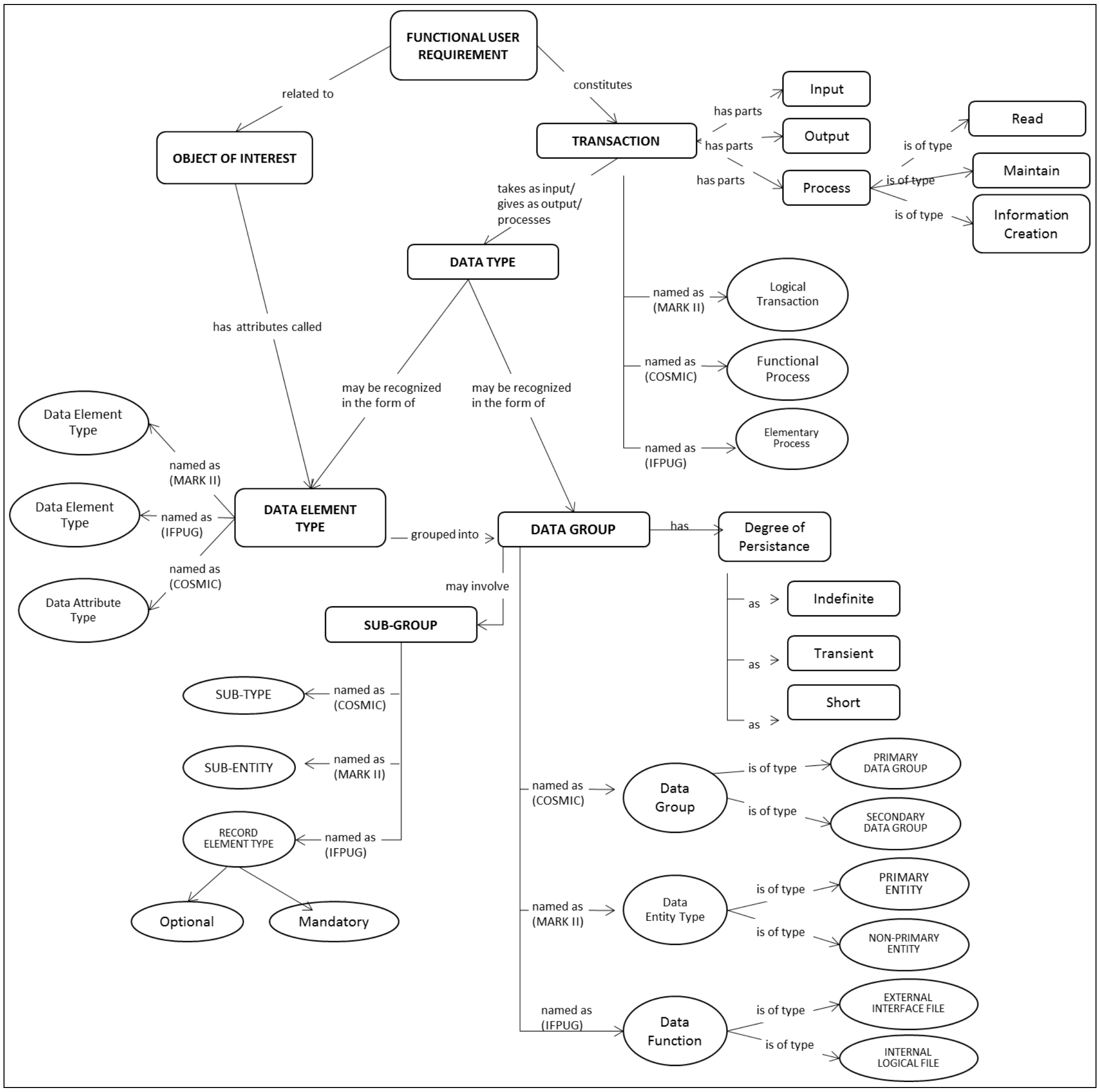


Figure 1.3: Conceptual Associations between FSM Methods-adapted from [3]

executed all that is required to be done in response to the triggering event. In FSM methods' models, transactional concepts are further characterized with respect to their associations to several data concepts of the models.

Data Concepts. FSM models define various data concepts (or data types) that represent the information maintained by the software and exchanged with its interacting users. Typically, the models define data elements as the smallest unit and define further data concepts that comprises logically grouped data elements as constituent parts of their models. Accordingly, IFPUG defines a data element type (DET) as a unique, user recognizable, non-repeated attribute and defines a Logical File (LF) as the user recognizable group of logically related data or control information. Record entity type (RET) is defined as user recognizable sub-group of data element types within a LF. Similarly, MARK II defines a data element as the unique user recognizable, non-recursive item of information and defines Data Entity Type as a fundamental thing of relevance to the user, about which information is kept. COSMIC define a data attribute as the smallest parcel of information meaningful to a functional user. Object of Interests is defined as anything in the world of the functional user about which the software is required to process and/or store data. A data group is defined as a distinct, non empty, non ordered and non redundant set of data attributes where each data attribute describes an aspect of the same object of interest.

1.2 The Problem

The reliability of a measurement method refers to the consistency of a number of measurements obtained using the same measurement method on the same subject. If repeated measurements are highly consistent or even identical, then the measurement method has a high degree of reliability. If the variations among repeated measurements are large, then reliability is low[70]. Unreliable results may lead to poor estimates, planning or misleading decisions. The more sensitive the figures are to the variations in functional size, the higher the uncertainty and the risk. Subjective measurement results obtained by the application of a particular measurement method adversely affect the reliability of a method and limits the usability of obtained software size for comparison and benchmarking. Therefore, any inconsistency in the measurement results deserves attention. In this respect, the utility of a functional size measurement method is determined by the reliability of the measurement results obtained by applying it.

FSM research acknowledge that subjectivity is among the critical problems of FSM [28][71] and inconsistencies are observed between the functional size measurements performed by different measurers on the same piece of software and using the same method [20] [21] [23] [22] [24] [25] [26] [27] [29]. The empirical studies reported various dispersion figures for the measurement results obtained by using the same method on the same subject. In [20] Rudolph reported measurement values obtained by IFPUG to be in a ±30% range around the average. In a following experiment [21], Low and Jeffery reported a range consistent with Rudolph. Later, Kemerer found 26% mean pairwise inter-rater reliability between pairs of measurers that count the same application. In [29], a dispersion of ±33% and ±51% around the average was found for IFPUG and COSMIC methods respectively. The reliability research on FSM methods mostly focused on reproducability characteristics of the methods. The assessments of reproducability were limited to closeness between the reported numeric functional size values and few of them investigated the sources of discrepancies [25] [26] [27] [24] [72]. FSM literature addresses three major factors to cause inconsistencies in measurement results[73]. First factor refers to the level of measurer's experience and knowledge in the selected FSM method and the software domain of the measurand. Second is the clarity and level of detail in functional user requirements and their suitability for FSM. Third is the different interpretations of the FSM method concepts and rules.

In this thesis work our focus is on the third factor. When a FSM method allows a room for different interpretations of the constructs and rules of the method's meta-model, then the consequences are discrepant functionality models of the same software constructed by different measurers. While this situation constitutes a shortcoming for any measurement method, the problem grows bigger when the validity of interpretations can not be decided according to the method's definitions and rules and the impact of interpretations are critical[26] [27]. Therefore methods need to provide a precise characterization of software *functionality* attribute to be measured and its counterpart components in its meta-model, while preserving its robustness against variations in the software functional requirements.

1.3 The Solution Approach

One approach for resolving ambiguities in the understanding and identification of the FSM meta-model concepts is formalization. Formalized concepts and rules enable consistent and precise definitions. The proposed formalizations in FSM literature (Chapter 2) roughly fall into two categories that are distinguished by their purposes. The first set of proposals address the mapping of some formal or semi-formal elements of requirements specifications to selected FSM method constructs and attempt to enable consistent interpretations by offering a set of formal mappings between the elements and constructs. Second category of proposals give formal general presentations of FSM methods based on similarities between meta-model constructs and analyses several properties of the methods and explores opportunities for conversions between models. None of these studies provide a concrete and explicit proposal that characterize FSM method model concepts and their relations. In this thesis study, we propose a formal software functionality model for FSM (4FSM) in order to facilitate precise and consistent measurements that result in reliable functional size values. 4FSM is based on the idea that events in the environment in which software operates are responded by an appropriate software behavior such that the environment behavior is satisfied. Accordingly, in our model software functionality is expressed in terms of software behavior and requirement specifications are taken as software artifacts that describe software behavior. 4FSM follows a stimulus-response view of software behavior where a stimulus is an event at the software interface and the response is the consequent behavior defined by requirement specifications (Fig 1.4). We chunk software behavior in terms of stimulus-response(s-r) pairs such that events in the world external to software are responded properly by execution of one or more s-r pairs determined by the state of the environment and software at the instance the events happen. In the model we give formal properties of s-r pairs, describe their dependency relations and give associated rules. A formalized measurement procedure is also given so that the dependencies between s-r pairs are handled, and software behavior is normalized into form suitable for quantification. In the measurable form each s-r pair exhibits a specific software behavior that can externally be observed and s-r pairs can be identified in homogeneous units which are characterized by their conformance to the the same set of given properties. Stimulus-response pairs have certain associations with the information software maintains and exchanges with its functional users. In 4FSM, the information is categorized into two types: data groups and control commands. Control commands comprise a set of parameters and are send by software to control the environment via functional users or send to software by its functional users to execute s-r pairs. Data groups are the homogeneous sets of data attributes of the elements of the subject domain structured with respect to the relation model of data [74]. We allow flat and nested relation representation styles so that the various structures of data that exist in the functional users' world can be represented while preserving homogeneity. We further distinguish data with respect to internal and external presentations of data and give associated properties in their representation. Internal presentation (Surrogate Data Model) relates to data groups that is maintained inside software boundary and determines the observable state of the software. External presentation relates to data groups exchanged between software and functional users.

A model with well-founded constructs coupled with formal presentations enables a consistent interpretation and understanding of the meta-model (Step-1 in section 1.1.2) and equip mea-

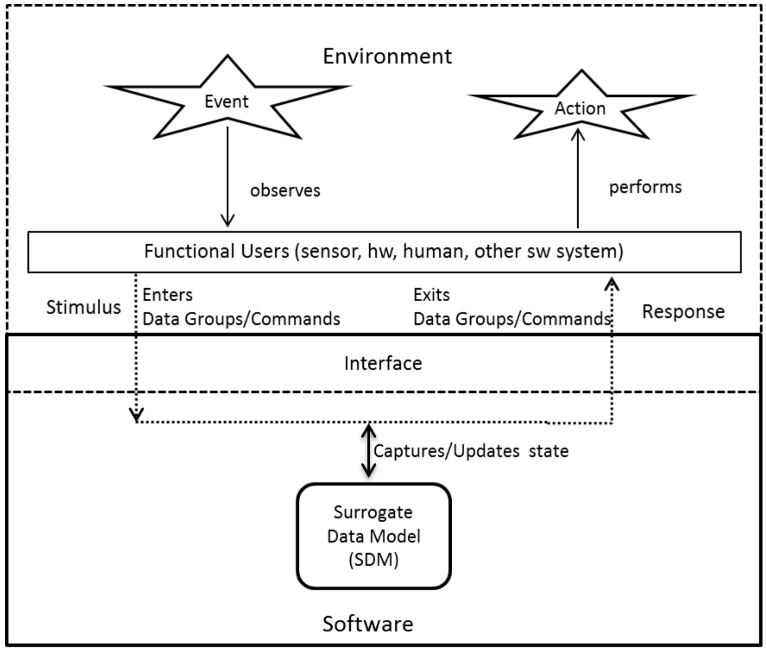


Figure 1.4: 4FSM Stimulus-Response Behavior

surers with an objective and precise basis to be used while constructing and verifying their models(Step-2 in section 1.1.2). Thus, 4FSM is expected to improve the overall software measurement performance enabling consistent and reliable measurement results.

1.4 The Research Strategy

In order to identify the sources of discrepancies resulting in inconsistent functional size measurement results and provide input for 4FSM model design process, we conducted two exploratory multiple case studies. They both had COSMIC measurement results as their subjects of analysis. In both studies, the analysis were performed were not only based on numeric size values obtained but also on the functionality models constructed by measurers. The first exploratory case study had measurement subjects from twelve software products from five different organizations. Our aim was to identify patterns in requirement specifications which are prone to inconsistencies in the measurement results in real-life cases. In the second study, we have conducted the case in a defined context where the specifications included properties that we have found to explain discrepancies in the first case and several other patterns that are reported in FSM method guidelines and have potential to cause discrepancies [75][54].

Our aim was to validate the error patterns, their relation to selected specification properties and identify the reliability ranges for COSMIC measurement results. We also identified the constructs of the FSM method meta-model that relate to measurement challenges. We took the findings and results of the exploratory case studies into account in the design of the 4FSM model. We performed a systematic literature review of formalization studies in FSM in order to understand the extend and limitations of the proposals and evaluate whether they contribute to the reliability problem that we defined. Besides, we have explored formalism approaches to be used in modeling software functionality. We have evaluated the suitability of the application of graph, set and logic theories. We have analyzed the properties of formal specification languages such as Z[76] , VDM [77], Event-B [78] and process algebras such as CSP [79], CCS [80] and various event-algebras [81][82] for expressing a generic model of software functionality. Besides, we have reviewed requirements engineering literature for a proper and clear definition of functionality and selection of software artifacts that describe software functionality. In order to validate the model in a measurement process first we mapped 4FSM and COSMIC FSM model concepts with the aim of showing how 4FSM model can be effectively used in an ISO FSM method. Then we performed another set of two case studies for the validation of 4FSM. In the first validation study, we applied 4FSM model to a requirement specification set and retrospectively compare it to previously obtained COSMIC measurement results. In the study we investigate how consistency could be improved in a measurement process with 4FSM and we address how 4FSM model handles several challenges encountered in previous measurements of the same set of specifications. In the second validation study, we conduct a multiple-case study with subjects of eight measurement results obtained from the same requirement set. We compare the consistency of the 4FSM models to the COSMIC models of a similar set of specifications. We also get feedback from the measurers on the perceived characteristics of 4FSM model.

1.5 The Organization of the Thesis

The remainder of the thesis is structured into four chapters. In Chapter II, the results of the literature review on formalization studies in FSM are presented and discussed. Also,a clarification on the fundamental concepts of Functional Size Measurement and a background on relational data modeling and stimulus-response behavior is given.

In Chapter III, 4FSM model is presented, the model constructs and their relations are defined and a mapping of COSMIC and 4FSM model concepts is given.

In Chapter IV, details for the two exploratory multiple-case studies and two validation case-studies conducted in the preparation of thesis study are given and their results are discussed.

In Chapter V, the contributions of this research to the field of software engineering and directions for future work are discussed.

CHAPTER 2

RELATED RESEARCH and BACKGROUND FOR 4FSM MODEL

In Section 2.1 FSM literature that introduce formalism into FSM methods and practices is presented. In Section 2.2 the background for 4FSM is given.

2.1 A Systematic Literature Review of Formalization Studies in FSM

This section summarizes the literature that add formalism into FSM methods and practices. The purpose of the review is to have an understanding the particular problems addressed by the proposals, the responses to the problems and their extend.

Section 2.1.1 explains the paper selection process and threats to validity. In section 2.1.2 we give an overview of papers grouped into two categories distinguished by their purposes. In the final section we summarize how formalizations in FSM help FSM practices and conclude on the results of the survey.

2.1.1 Review Method

The main criterion for including a paper in our review was that the paper should describe research on software functional size measurement and it should follow a formalization approach in the proposals to FSM problems. We limited the extent of the formalization approach in FSM research to defining models, notations or semantics for describing FSM methods and measurements. We included academic publications including, journals, books, conference proceedings, technical reports and MSc and PhD thesis. The proposals in the papers were expected to conform to FSM method definitions and rules without alterations or extensions to the methods. The papers that were a continuation of a previous formalization work were also included in the review. The papers only written in English were included in the review. Some papers in French were understood to present relevant work or were referenced by other reviewed papers [83][84].

2.1.1.1 Identification of Papers

First, we made a search on our paper collection that contains over 400 publications on FSM by reading the paper titles and abstracts. Initially a set of six relevant papers were found to satisfy the inclusion criteria [85] [86][69] [87][88][89].

Then, the selected papers were fully read and other research materials that were referenced by the papers were found. When a referenced paper matched the inclusion criteria, it was added to the set of identified papers. The same procedure was repeated for every paper added to the set. Additionally, FSM methods' official websites were checked for other relevant studies. The journal, conference and institutions that published the selected papers were identified and other papers were found through a keyword search in relevant publisher or organization sites and academic digital libraries. Additionally, candidate papers were found by keyword searches on the academic search and citation search engines. Finally, five journal [90][4][87][91][89] nine conference papers [92][86][85][93][69][94][95][96][88] and four technical reports [97][98][99][100] were identified to satisfy the inclusion criteria.

2.1.1.2 Threads to Validity

Publication Bias. As the discussions and their summaries in the selected papers indicate, the research materials that were written in languages other than English were likely to be included and analyzed in this review. However, when the discussions and the summaries of those papers in English are assumed to be representative of the formalization ideas explained therein, they were not evaluated to change the results of our review critically. Formalization in FSM is a subject that can potentially take place in a variety of software engineering publications and conferences. The sources to search for related papers were numerous. For this reason, some relevant works may have been overlooked and this may affect the quality of the analysis negatively.

Anachronism. The time span of the papers range from 1991- 2009. In this period, FSM methods were introduced and some have gone through modifications that have been released as method versions. Furthermore, software engineering has been experiencing paradigm shifts in this period. Although the underlying principles of FSM methods have remained almost the same, some formalization suggestions may not be valid or may be obsolete today. Our analysis and observations were based on these principles and core definitions described in of each FSM, which we believe to have been valid in 1990s and today.

Limited Scope. All of the papers we analyzed proposed formalizations in one or more of IFPUG, COSMIC and MARK II FSM methods. The results of our analysis and observations may not be generalized to all FSM methods.

2.1.2 Overview of the Papers and Their Discussion

The papers were classified into two categories, where the categories were identified by a clear separation of the purposes of suggested formalizations. The first category papers have explained how added formalisms in FSM can contribute to measuring software functionality described in a specific language, notation or model. The category was named as ”Formalizations in measurements from formal specifications”. The second category papers have proposed formal presentations for FSM method concepts and measurement functions. The second category was named as ”Formalizations in FSM foundations”. Although they cannot be perfectly isolated from each other, the motivations for all papers can be given as:

- resolving ambiguities in FSM method concepts and rules due to abstract or insufficient definitions,
- decreasing or explaining variability in size of the same set of FURs due to different interpretation of FSM concepts and rules,
- decreasing measurement errors and
- exploring the steps of measurement for automation possibilities.

2.1.2.1 Formalizations in Measurements from Formal Specifications

The formal languages describe the desired software functionality in terms of a set of specification elements. FSM methods describe functional aspects of software in terms of the functional components in the abstract software model of a FSM. As their common characteristics, papers of this category establish correspondences between the functional components of a FSM method software model and the elements of the specification language so that the functional size can be measured directly from the specifications. The correspondences are based on a set of rules that are defined in each specification's context; such that the counting rules and functional components are interpreted, redefined and presented in selected specification notation terms. Thus, the papers propose the automation of this essential step of concept mapping in the FSM process by introducing an automated measurement process for a selected language via formal rules, eliminating the manual work.

One set of studies proposed procedures to measure functional size from requirements in a formal specification language. In [97], a classification of formal specification languages and an evaluation of the adequacy of language classes by looking at their suitability for formalizations of IFPUG concepts is given. As a continuation of this work, Diab et.al. performed a syntactic analysis of B specification language and the correspondences between IFPUG and B specification concepts and the formal rules for identification of IFPUG functional components for B were given [86]. In the study, IFPUG method completeness was discussed based on the cases that can be specified in B but are not covered in IFPUG. They explained how added formalisms could be used in the structural analysis of the IFPUG method model by showing examples of the concepts that need human judgment for identification and can cause subjective results. In [85], the same approach was followed for ROOM (Real-Time Object-Oriented Modeling) specifications and COSMIC-FFP FSM method. They interpreted COSMIC concepts and rules and then defined formal rules in first order logic and set theory to be used in functional component identification. They also found that there is not a corresponding specification concept to map to the layer concept in COSMIC software model. Later in [91], they implemented the rules to support automatic measurements from ROOM specifications in a tool to run in a RRRT (Rational Rose RealTime) development environment. In a case study, they compared manual measurement results of a COSMIC expert to the results from the tool obtained automatically and analyzed the sources for variations. Similarly, Miyawaki et al. proposed a method to measure IFPUG FP from specifications in Vienna Development Method- Specification Language (VDM-ML) [93]. They interpreted IFPUG concepts and presented the rules that map the VDM-ML concepts to IFPUG functional components in a mathematical notation. They implemented mapping rules in a tool; they compared manual and automated measurements in a case study and analyzed the sources for variations.

The major motivation of this set of papers is automating functional size measurement from specifications in a state based formal language as classified in [97]. The formal rules added to this automated process are expected to yield consistent results when considered with the concept interpretations that constitute a base for the formalized rules. Commonly, the papers explained the openness of FSM functional component definitions to interpretations and then justified their reasoning in their concept mappings.

Another group of studies defined IFPUG measurement procedures applicable to specifications given in Data Flow Diagrams (DFD) and Entity Relationship(ER) models that are mostly used in structured development environments. In his proposal, Rask established the conceptual correspondences between the DFD elements and IFPUG concepts [100]. The method included a series of algorithms applied to the specification to identify the IFPUG functional components. The algorithm statements were based on DFD and ER diagram notation elements. Later, Rask used the algorithms in a simulation study that compares function points with another function metrics [90]. Gramantieri et al. followed the same approach [98]. However, they replaced DFD data stores with E-R entities and ER relations, thus integrated DFD and ER concepts. They translated conditions that are handled by IFPUG rules into formal rules in terms of properties of the ER-DFD graph and then implemented the rules in Prolog logical programming language. As a precondition for defining formal rules, a set of assumptions were made to enable consistent interpretations of IFPUG concepts in ER-DFD specifications. In [4] this study was extended with case studies and they obtained close results by automated and manual measurements. Figure 2.1 shows the ER-DFD graph properties on an example and an ER-DFD instance for a requirement from a series of case studies published by IFPUG [101]. In the measurement procedure they propose, the software model elements such as elementary processes, Logical File (LF), Record Entity Type (RET) and Data Element Type (DET) are identified in the graph running the interpreted rules. Note that the formalism in our focus is not the specification of the requirements in a formal language but in translating the informal counting rules expressed in natural language in the IFPUG manual into rigorous rules expressing properties of the ER-DFD graph. Once the rules are given in terms of directed arcs, dashed lines, ER multiplicity symbols and other elements, it was possible to automate measurements sticking to the set of assumptions and rule interpretations.

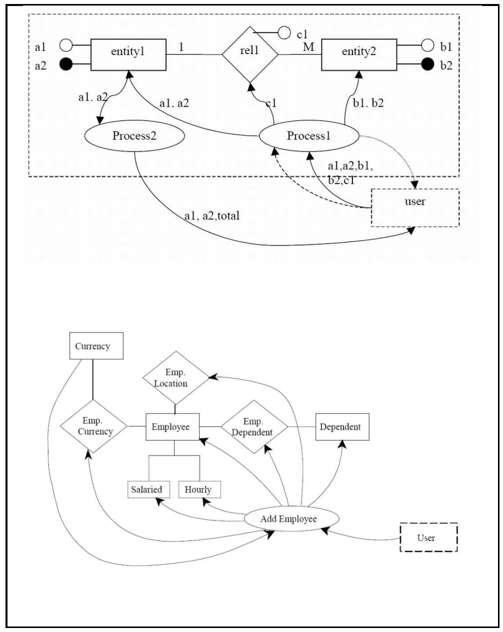


Figure 2.1: ER-DFD Graph Notation and an Instance-adapted from [4]

In [99], Abran and Paton used a DFD like formal notation in presenting the rules for IFPUG with the motivation of exploring the measurement activities that can be automated in IFPUG. The notation consisted of graphical symbols that represent processes, stored data, data manipulations and software boundary. They used this notation and evaluated all possible patterns for a given process against IFPUG process type (EI, EO, EQ) identification rules. They identified the patterns which require human involvement in determination of the corresponding process type and which do not. Then, they proposed an extended notation to include data files, record types and data fields and defined a measurement procedure for specifications in the given notation. Later, April et.al extended this notation and represented formal rules that associate IFPUG model to computational concepts such as source code, user interfaces, and software modules [92]. They explored the use of these rules in calculating functional size from source code as part of a reverse engineering technique.

This group of studies show similar characteristics in their formalization approach with the papers that introduces formalizations to measurements from state based formalization languages such that the applicability of the rules depend on assumptions or interpretations on

IFPUG counting and identification rule and the specification language. This group of studies suffered fewer difficulties while fitting the software specified in ER and DFD into IFPUG meta-model, since FSM method models are data-oriented and data analysis terminology is used in concept definitions, rules and examples in the IFPUG CPM manual [54].IFPUG manual includes sections that guides the identification of ILF, EIF and RET data types via rules in ER relation notation in parallel to this approach these studies share; however, they do not perfectly match since they give rules in different sets of ER properties and only data functions are in its scope.

In summary, the papers of this category provided means and explored automation opportunities to measure functional size for software described in the specification language in their scope and the associated FSM method. However, a functional size measurer should carefully consider the concerns discussed in this section before attempting to use the ideas in practice. Table 2.1 gives a summary of papers where FSM method concepts and rules were redefined formally to enable automated measurements from a specification language.

Table 2.1: Formalizations in Measurements from Software Description Styles

Research References		Specification Language	FSM Method
Original Study	Continuation Study		
[85]	[91]	ROOM	COSMIC
[100]	[90]	ER,DFD	IFPUG
[98]	[4]	ER+DFD	IFPUG
[86]		B	IFPUG
[93]		VDM-SL	IFPUG

Our final observation on the papers of this category is that the analysis of FSM method concepts from a formalization point of view contributes to a more clear identification of the ambiguous points in method definitions and rules. The analysis of the FSM concepts while mapping the specification language elements reveals possible sources for subjective measurements. Moreover, the analysis results addresses the points that need human intervention in an automated measurement process, thus supports automation tool design.

2.1.2.2 Formalizations in FSM Foundations

The papers in this category propose formalizations in representing the functional components defined in FSM method abstract models and the measurement functions. The purposes of the studies are providing formal FSM model definitions in the solution of problems resulting from ambiguities in FSM method concepts and rules. The purposes also include the investigation of the differences in FSM method structures, revealing opportunities for method improvements and automated measurement; hence promoting a better understanding of FSM methods.

In [69], Fetcke introduced the idea of defining a generalized structure for IFPUG FPA and MARK II FPA and COSMIC FFP as FPA variants. In the study, two steps of data oriented abstractions in FPA were identified: software requirements are represented in data oriented abstraction (identification of items) and the items in the data oriented representations are mapped into numbers (mapping into numbers). He introduced activity type concept so as to represent the different concepts besides the common concepts in FPA and its variants. Then, the abstractions were formalized in a mathematical model. Using this model, he formally presented and tested empirical assumptions of dominance and monotonicity which are made by Function Point Analysis. In a subsequent study, Fetcke extended the original work and validated the completeness of the generalized presentation against the concept identification rules in different versions of three FSM methods; COSMIC FFP, IFPUG and MARK II [94]. In this continuation study, he added Control activity type, to enable a better representation of IFPUG and MARK II concepts in the model. Finally, he explored the potential applications of the generalized formal model. The generalized model is shown in Figure 2.2. In the figure, the abstractions from software documentation to data oriented software model and from model elements to numbers are illustrated. The core concepts of user, application, data and transaction are shown. The definitions of the model are given in Table 2.2.

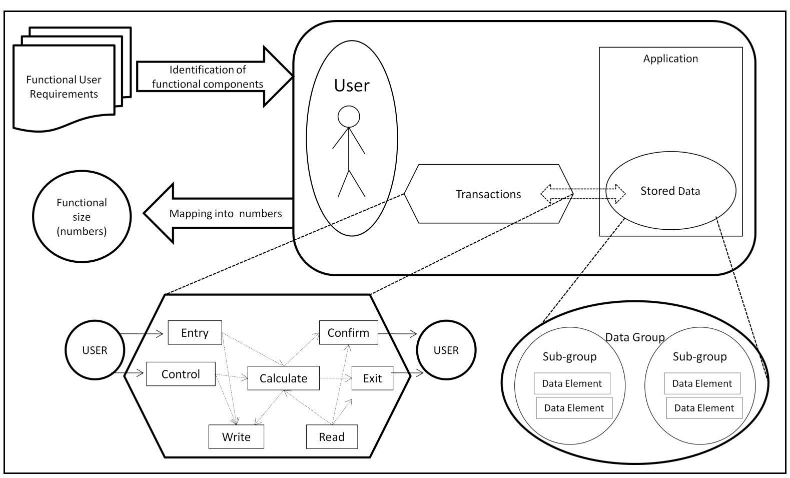


Figure 2.2: Generalized Model for FSM Methods

Table 2.2: Generalized Formal Representation of FSM Methods

Definition	Formal Presentation
Application closure is a vector of τ transaction types T_i and σ data group types F_σ.	$H = (T_1, \ldots, T_\tau, F_1, \ldots, F_\sigma)$
Activity P_{ik} is a quadruplet ,where θ denotes activity class, $\theta \in \{Entry, Exit, Confirm, Read, Write, Calculate\}$ r denotes data group type referenced, D denotes set of data elements handled , C set of data elements calculated for Calculate activities.	$P_{ik} = (\theta, r, D, C)$
Transaction type T_i is a vector of activities	$T_i = (P_{i1}, \ldots, P_{in})$
F_j is a set where, d_{jk} are data elements g_{jk} are designate data sub-groups	$F_j = \{(d_{j1}, g_{j1}), \ldots, (d_{jn}, g_{jm})\}$

Later, Hericko et al. proposed a measurement model that includes steps of converting any software model into a universal model that is based on Fetcke's generalized representation and measuring the functional size in any of the three FSM methods [87]. In their study, they presented instances of Fetcke's generalized presentation for the latest versions of the three FSM methods; IFPUG, COSMIC and MARK II FSM. They formally represented measurement functions in the notation developed by Fetcke and they defined method executions in symbolic code. They showed an execution of the presented measurement process for Object Oriented specifications; they mapped Unified Model Language (UML) elements to the universal model elements in symbolic notation and then formally modeled the software functionality in Fetcke's general presentation.

In (Hericzko, 2006) software description language elements for UML were mapped to functional components of the generalized model instance for IFPUG, where data groups and subgroups were identified via a set of interpretations similar to direct language-FSM model element mappings such as given in [102]. Hence, the aforementioned gap problem remains to be inherited in performing measurements with respect to the generic model. The generalized data oriented abstract model, providing formal presentations, contributes to understanding the differences between the functional components in each method model and provides a tool to discuss the formal foundations FSM methods are based on.

Demirors and Gencel defined a unified software model that is a resultant set of IFPUG, MARK II and COSMIC FSM method software models[89]. In order to construct the model, they first performed a conceptual analysis of each method model and identified the commonalities and the differences between the model concepts and developed a conceptual map of method concepts [88].Then, they defined the unified model elements, which constitute a superset of all concepts required by each model, and they gave rules that assign model concepts in terms of unified model elements via set and assignment operators. Most uses of the unified model were considered to be in simultaneous measurements and conversion of sizes measured by different methods. The unified model was implemented in an automated measurement tool and it was reported that the measurement results obtained following the unified model and application of each individual method were consistent in several case studies. The identification of the components in the unified model requires the knowledge of the concepts and rules of each supported FSM method. For example, following the mappings of a data group and unified model rules, a measurer may infer that a data group in COSMIC method corresponds to a data group in IFPUG.

Bevo, Levesque and Meunier proposed an ontological formalization for a software functional size measurement method's application process[96][95]. Their work highlighted the problems of technical difficulties, lack of automated support in applying FSM, and addressed the essence for domain and task ontologies in an FSM procedure. It was explained that the proposed ontological formalism contribute to a better understanding of the measurement procedures of related methods, and facilitated the understanding of concepts for structuring, representing exchanging and interpreting information related to the measurement procedure. Among the several formalisms to present ontologies, object-oriented formalisms were justified and selected. Their works provides instances of domain and task ontologies for IFPUG, COSMIC and MARK II FSM methods and puts all method definitions and concepts into a schema and shows the relations among them.

The second category papers were mainly concerned with FSM methods' abstract models and added formalizations into presentations of the models, model elements and their relations so that the FSM methods and measurement functions are understood better.

2.1.3 Summary and Discussion of the Related Research

After an analysis of the papers included in our review, we concluded that formalization studies in FSM help FSM research and measurement practices in three typical ways:

1. Exploring opportunities for automation: The measurement steps and concepts that can be formalized are good candidates for automation. The steps that can be fully automated, semi-automated or cannot be automated are identified.

2. Automation of measurement through consistent interpretations: When the formal rules for identification of the FSM method model concepts from a specification form are given, since they are repeatable, they can be interpreted consistently for every piece of software described in the same form. The mapping phase in measurement is automated, producing results that conform to the method definitions and rules.

3. Understanding FSM model structures: The formal definitions of method models and measurement functions enable method comparisons and determination of the suitability to the measurement purposes, thus describes the similarities and differences between the models transparently.

One observation we had throughout the study was that the efficacy of the use of formalisms was determined by the degree that the abstract software model of a FSM method and its rules can be presented formally. The measurements are open to subjective results, when abstract software model elements are interpreted for a requirement description language. All reviewed papers and many others in FSM literature, consensually agree that in practice, reasoning that the method concept and rule descriptions given in the manuals and guides are too abstract to be applied directly to software specified in a selected language, FSM methods need interpretations. Without interpretations, the required transformation of software descriptions in a language to the abstract measurement method model can potentially result in inaccurate, inconsistent and imprecise measurement results. A variety of FSM research was made to fill this ”gap” between software description forms and the FSM software abstract models. They came up with proposals that interpret or specialize the functional concepts and measurement rules of a selected FSM method systematically for a selected software description form [103] [104] [105] [102]. A recent study reviews the procedure developments for sizing software with COSMIC FSM method [106].

As the FURS can exist in any specification language, the measurer's abstraction approach and interpretation of the functional concepts and rules in a FSM method may lead to different results obtained by different measurers, regardless of the formalism in these interpretations. The formalization studies in the first category proposed partial solutions to this problem by representing the authors' interpretations for a formal language formally. Thus, the interpretations became repeatable and consistent for measurements of the software described in the same language, the consistency in the results is achieved and automations were possible. Nevertheless, the formalizations do not significantly change the nature of the subjective measurements problem due to different interpretations. When the interpretations are different, although they are formally represented, they may still result in different FSM models for the same piece of software. For example, although a DFD process may not always correspond to an IFPUG elementary process, given a set of assumptions, constraints and through formalizations, they may be consistently interpreted so. Furthermore, although consistent interpretations followed in the papers can provide comparable measurements results, they localize the standard methods, thus, it may lead to incomparable software functional sizes for benchmarking purposes or obtained from other specification styles. The studies in the first category papers in our review partially eliminated the potential problem getting contributions from FSM method founders or co-founders. Nevertheless, it becomes a problem when FSM method experts and founders are scarce and there exists a variety of software description languages that specify software functionality.

The inherited problems with FSM method definitions and structures, which turned out to be the weaknesses of a formalized measurement process, were clearly identified in the formalization processes of the first category papers. It was not a coincidence that these formalisms were for software descriptions in formal state based specification languages (B,ROOM,VDM-SL) or semi-formal ER, DFD presentations. Since their major motivation is automation of FSM, more formal descriptions of software increased the opportunities for automated measurements and delegated the problems from ambiguous requirements in FSM to the requirements engineering domain. The effectiveness of formalizations were tested by few cases where some test or synthetic cases reported by the paper authors' were limited to the analysis of case specific variations between manual and automated measurements and few of the case studies were conducted in an industrial setting. Most of the mappings in this category of papers are based on older method versions, hence may need revisions to adapt to latest versions of the FSM methods.

The generalized software abstract model explained and presented in Fetcke's work and used in measurement process Hericko et al. was structured to include the relevant information that is deemed relevant to software functional components for all three FSM methods. Nevertheless, the formal model presentations do not include any abstractions for component identification rules which describe the qualifications expected from a valid functional component in the context of a selected FSM method. A data group in IFPUG may not be valid in COSMIC or vice versa. The data group and sub-group concepts in Fetcke's model are also abstract and they become concrete and valid only in a method's context conforming method definitions and rules. For example, in COSMIC FSM manual [8], one data group identification rule is stated as "Each data group shall be directly related to one object of interest in the software's Functional User Requirements". It is obvious that accurate identification of objects of interests is required in order to validate an identified data group; the inaccuracies will directly affect the measurement results.

In [89], Demirors and Gencel give associations between method concepts and however it does not unify the rules which are valid in each method's context. Without knowing each method's details, the exact and complete identification of the concepts in the model is not possible; so the unified model should rather be regarded as a simultaneous measurement method that approximates and unifies the methods with a focus on the similarities in method concepts. However, following the unified model, measurement effort can be saved from eliminating the rework for identification and measurement of common concepts.

FSM method software models are data oriented and the data requirements described by the functional user should be carefully analyzed to measure the software functionality accurately. In IFPUG CPM Manual Part 2, data modeling concepts were introduced and the mapping of ER concepts to IFPUG LF, RET and DET was given in a formalism that is represented in Entity-Relationship properties of multiplicity, participation and entity dependency types. Although ER formalism may not be sufficient or suitable to reflect all aspects of data related concepts as perceived by a FSM method, when the method rules are described formally and are based on a formal data model, measurers will be supported in identification of the method data concepts from the FURs, where user data is described. For example, COSMIC FSM business application guide explain functional component identification rules referring to ER formalism in order to clarify the ambiguities in the relations between functional components where the same information can hardly be produced from COSMIC manual [75].

As the final conclusion, we believe one significant and common contribution of the all formalization studies came from the formalization processes themselves. A set of critical problems with FSM have been discovered, revealed and communicated in concrete terms. Besides,the researchers were equipped with a basis to be used in evaluating and expressing FSM method improvement suggestions.

2.2 Background for 4FSM

Section 2.2.1 clarifies 4FSM view of software functionality and associated software artifacts. 4FSM model relies on stimulus-response behavior and relational model of data. For the sake of completeness, in Section 2.2 an insight into stimulus-response behavior of software is given and in Section 2.2.3 basic definitions of relational model of data are summarized.

2.2.1 Software Requirements, Specifications and Functionality

Ambiguous concepts brings in serious methodological risks and clear definitions of fundamental software concepts,the entities and their attributes to be measured is required before defining any measure or measurement method[107]. Therefore, it is essential that we first discuss and clarify the fundamental concepts our model relies on; such as software requirements and functionality.

ISO/IEC 14143-1 defines Functional Size: ”a size of the software derived by quantifying the Functional User Requirements”. There are many different views on what a software requirement is. In practice, they are used to refer to the properties of the environment where there exists a problem or the desired properties of a solution system which is to operate in the environment; or it used to address technical and business constraints and even for the budget or schedule objectives pertaining to a software project. ISO-Systems and Software Engineering-Vocabulary [108] defines a requirement as:

> (1) a condition or capability needed by a user to solve a problem or achieve an objective
>
> (2) a condition or capability that must be met or possessed by a system, system component, product, or service to satisfy an agreement, standard, specification, or other formally imposed documents.
>
> a documented representation of a condition or capability as in (1) or (2)

This definition covers a broad range of concepts that constitute different types of concerns. Nevertheless, the clarification and separation of these different concepts are vital for functional size measurement as well as successful management of software projects. The clarification of the requirements concepts has been one of the major achievements of requirements engineering research and a significant portion of contribution came from M.Jackson (with others, notably Pamela Zave) throughout problem analysis studies(also known as problem frames approach) [109][110]. Accordingly, software artifacts are studied in a universe of two connected domains where problem domain is the part of the universe where the problem exist and the solution domain (or machine) is the software-intensive system. A part of the real world becomes a problematic environment because its current behavior is unsatisfactory in someway. The developers solve the problem by building a software-intensive system and connecting it to the environment in such a way that the behavior of the environment becomes satisfactory (Fig 2.3). In this view, requirements are the optative statements about the environment and they describe the desired effects to be brought about in the environment by the behavior of the solution system. Requirements do not directly concern the machine and they can be stated without reference to the solution system. There exists some shared phenomena

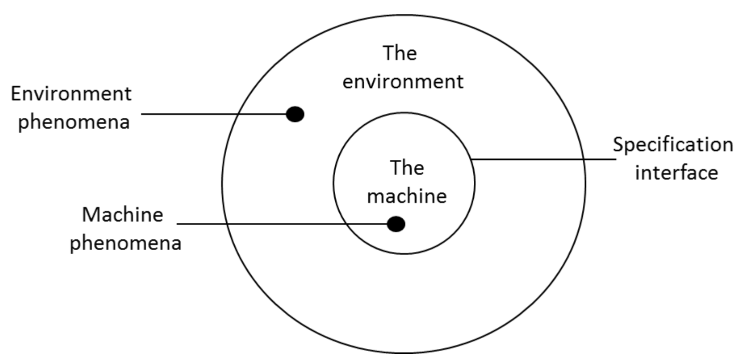


Figure 2.3: Software Specification

common to both the environment and the solution such that software system can affect and be affected by the environment. These are the events and states that are observed both by the environment and by the solution. Shared phenomena are either controlled by the solution system or by the environment. The shared phenomena are named as the specifications which connects the two domains and constitute the interface between the two domains where interactions occur. Thus, specifications are the desired properties of an observable behavior of a solution system defined such that the solution system produces the desired effects in the environment [111].

A more formal view of this model is given as a reference model described in terms of five software artifacts [5],(Fig. 2.4). Domain knowledge (W) provides presumed facts about the

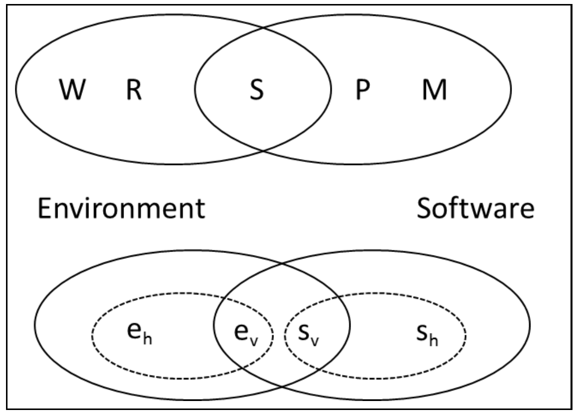


Figure 2.4: Software Artifacts-adapted from [5]

environment; they are the indicative properties of the environment regardless of the behavior of the software. Requirements (R) are desired properties expressed in terms of its effect on the environment. Specifications (S) are the shared phenomena that connects the software system to the environment. Program (P) implements the specification using the programming platform; and a programming platform (M) provides the basis for programming a system that satisfies the requirements and specifications.

Phenomena controlled by the environment and system are denoted as e and s and are further partitioned with respect to their visibility. Thus, $e = e_h \cup e_v$ and $e_h \cap e_v = \emptyset$. Similarly, $s = s_h \cup s_v$ and $s_h \cap s_v = \emptyset$. At the interface(S), the shared phenomena are denoted as e_v and s_v hence they are visible to system and environment, respectively. Their hidden complements in each domain are denoted as e_h and s_h and are private to the domain that controls them. The Venn diagram at the bottom of the figure shows the relationships among the four sets of phenomena.

A tiny example illustrates the distinction between the four phenomena: The requirement R is that security staff wants to be notified if there is a fire at a floor when they are in the security room. There is a programming platform M with a smoke sensor to detect fire at a floor and an actuator that can be programmed P to sound a buzzer based on data received from the sensor.

Domain knowledge says that there is always an officer in the security room to hear the buzzer, and if there is smoke then there is fire. The terminology falls into four groups (Fig. 2.4):

e_h: the security officer and the fire

e_v: smoke from fire

s_v: the buzzer at the security room

s_h: internal representation of signal data from the sensor

The specification S, which is expressed in the language common to the environment and system, says that if the sensor detects smoke then the system should sound the buzzer. Accordingly, while a specification(S) describes machine behavior sufficient to achieve the requirements, S must have a vocabulary common to the environment and the system. In other words, the variables of S must be among those in e_v and s_v and they cannot include any of those in e_h or s_h; s_h is phenomena of internal behavior which entirely depends on the configurations for P and M .

The transitive relation between the artifacts of the model relies on the following: *if S properly takes W into account in saying what is needed to obtain R, and P is an implementation of S for M, then P implements R as desired*. The model is accompanied with a set proof obligations in order to assure that a given set of specifications are consistent and comply with a given set of requirements. When such conditions are satisfied, there exists a specification (S) that describes an externally visible behavior s_v and its interactions with the environment (e_v) and that satisfies the requirements and a software system that is installed in the environment that enables the environment to behave as desired. The primary role of domain knowledge in requirements engineering is in supporting refinement of requirements to implementable specifications. Correct specifications, in conjunction with appropriate domain knowledge, imply the satisfaction of the requirements.

The separation and clarification of software artifacts as such provides us with several further conclusions that are vital for precise characterization of software artifacts to be measured in FSM. Requirements are the expressions of the desired behavior of the environment where specifications are requirements that are implementable. Thus, specifications form the vocabulary of interaction between the environment and the software and it should not include terms and concepts private to the solution. The former belongs entirely to internal design of the software. There may be more than one set of specifications that satisfy the requirements in the environment as there may be many ways to implement a working set of specifications. Thus, specification, as the activity of determination of an externally visible behavior of the software is an inventive process that takes domain knowledge and requirements as inputs [112]. The relation between visible software behavior and requirements can be complex and indirect such that the desired behavior of the environment upon the occurrence of an event private to the domain may involve software's behavior in a number of ways. For example as a consequence to various different events that happen in students' and researchers world', they desire to have a list of publications that meet several criteria with respect to the event. A specifications for a publisher database may define a behavior such that whenever several query criteria are selected and details are entered in a form, software returns all matching publications in a view (Fig 2.5). Thus, many different private events and associated requirements are satisfied by a single behavioral description. Specifications are the main instrument of communication

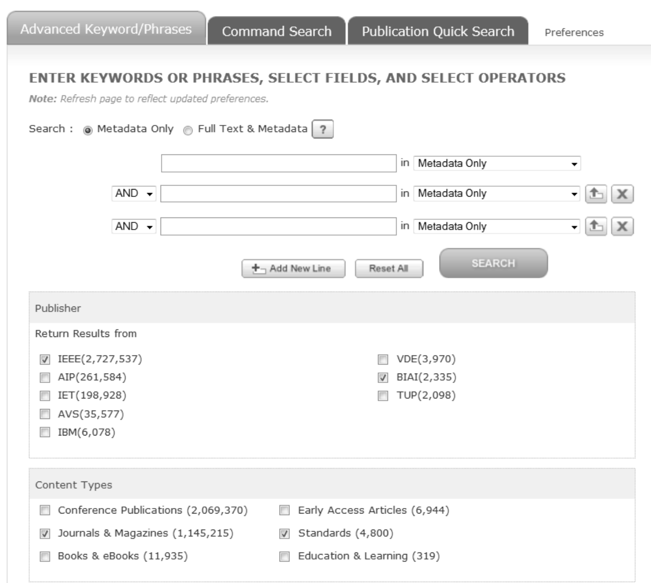


Figure 2.5: Query Publications Specification

between software users and developers.The availability, quality and documentation of specifications is vital in early project life-cycles. They identify the boundaries for developers' design freedom. However this does not imply that the specifications for designed software components do not exist. They are rarely available as application programming interface(API) or in some other format. Nevertheless, the all other software components and devices constitute the environment of a component and the externally visible behavior of software component can be described.

ISO 14143-1 does not specify any requirement, nor does it provide any directive or specification as to the characteristics comprising the concept of functionality. Some definitions of functionality from software engineering literature are as follows:

Functionality: Waffle for ”features” or ”function”. The capabilities or behaviors of a program, part of a program, or system, seen as the sum of its features” [113].

”Functionality captures an intuitive notion of the amount of function contained in a delivered product or in a description of how the product is supposed to be” [11].

”capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions.” [114]

Naturally, the usability and clarity of such definitions rely on what a ”function” is. Along with its pure mathematical meaning, there is a wide repertoire of software engineering concepts the term function refers to and which may involve the shared and private regions of the problem and solution domains or may pertain to a specific element software development or modeling technique (see [54] for a potpourri of definitions). It may refer to a defined objective or characteristic action of a system or component, or a software module performs a specific action. It may refer to boxes in a functional decomposition tree or the bubbles in a DFD as an abstract expression of software performed processes that transform inputs to outputs; or, it may refer to the purpose, role, goals, features of the software.

In an engineering context, a central meaning of function is ”function as desired effect” [115]. Accordingly, software is used because software users desire that something desirable happens outside the software and the desired effects can be stated entirely in terms of the behavior of environment (environment-centric). For example the function of the tiny fire-alarm software is ”providing a mean by which security officer in the security room is alerted that there is a fire in the building”. However,once the causal relations between the desired changes in the behavior of the environment and the external software behavior is established, the function can also be specified entirely in terms of the software as ”when smoke signal is received from the smoke sensor a signal is sent to the buzzer”. This view of functionality is consistent with the definition of requirements and notion of specifications as explained in the reference model. Therefore, the functionality can be stated in terms of the environment, software or a mixture of both. Inline with these definitions, for FSM purposes, provided that the connection between requirements and specifications are established, we are interested in both software and environment centric presentations which are essentially elements of a specifications .

Several taxonomies frequently used in practice classify requirements into categories of functional and non-functional requirements. The functional requirements capture the nature of the interaction between the component and its environment [116]. Non-functional requirements are considered to be related to the qualities the software behavior exhibits during its interactions with the environment such as responsiveness, security, portability and reliability requirements [58]. Design constraints are also considered as non-functional which can also be classified as technical requirements that restrict the potential set of solutions to the problem hence represent a condition that restricts the set of possible configurations for P and M of the reference model. While separation of requirements as such may have several benefits for different concerns of project management, requirements management, architectural or contractual decisions, this separation may not be useful or may cause ambiguity for some other purpose[117],[118]. ”Non-functional” requirements relate to software functionality and they may overlap with functional requirements and as such characteristics can be desired for the whole system, they can also be different for different interactions [112]. Furthermore, a ”non-functional” requirements can be precisely expressed as functional. For example, a security requirement may be fully expressed as a functional requirement and come to life as a login service or several failures may require error handling. A legacy architecture may mandate a query report to be prepared and served in pieces in a defined sequence or in an asynchronous fashion. However, due to their volatility or technical complexity, the relation between non-functional requirements and functionality may be apparent in later stages of development cycles [112]. To summarize, the overall software behavior is induced by functional requirements and whenever expressed as visible software behavior, quality requirements and design constraints may affect the eventual software functionality and software functional size [119].

The 4FSM software functionality model described in Chapter 3 is designed to be used in methods that attempt to measure functionality attribute of software. 4FSM relies on the definition of software functionality as the externally visible behavior of software and takes specifications as the artifacts that describe software functionality where the external manifestation is distinguished from the internal structure and internal behavior of the software.

2.2.2 Stimulus-Response Behavior

A fundamental concept we use is stimulus-response behavior which is used in modeling and decomposing software behavior and used mostly in reactive systems literature[120] [6]. A reactive software system is a system that, when switched on, is able to create desired effects in its environment by enabling, enforcing or preventing events in the environment.It is supposed to maintain a certain ongoing relationship with its environment. Analogous to biological systems, such systems respond to stimuli by possibly changing their state and influence their environment by sending back some signals to it (Fig. 2.6). Stimuli are the events that happen at the interface of the system. Responses are the messages and signals sent to the environment and they are computed as a function of the state of the system and environment. Events and the desired effects are communicated between the system and environment through a connection domain which includes entities that directly interface and interact with the system such as humans, sensors or actuators,etc. Reactive system behavior well represents a broad

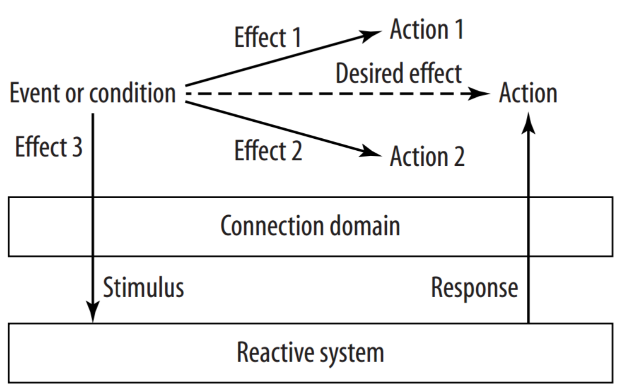


Figure 2.6: Stimulus-Response Behavior-adapted from [6]

category of systems including real-time and embedded systems. Also it extends to business applications such as information systems, work flow management or e-commerce systems [121] [6]. Reactive systems are often contrasted with transformational systems which lend themselves naturally to problems that can be decomposed and described uniformly in terms of inputs, outputs and a relation between them, such as algorithmic problems. The behavior of reactive systems depends on effects of the traces of past stimuli the system has responded hence states have an important notion in descriptions for reactive systems. Table 2.3 compares some characteristics of reactive systems to transformational systems.

Table 2.3: Characteristics of Reactive Systems

Reactive	Transformational
Highly interactive,prompted by the environment	May prompt the users to collect information about the environment
Interrupt-driven	Not Interrupt-driven
State-dependent response	Output do not depend on state, return the same outputs for the same input data.
Environment-oriented response	Output not defined in terms of environment
Usually, stringent real-time requirements	Usually, no stringent real-time requirements
Non-termination is desired	Termination is desired
Parallel processing	Sequential processing

2.2.3 Subject Domain and Relational Model of Data

Exchange of data (messaging) is the means of interaction between the software and its environment. The exchanged messages between the software and its users are about a part of the environment that is shared with software. This domain is called a subject domain and it forms a discourse between the software and its environment. The sum of subject domains of all possible messages exchanged by the software and its environment across various interfaces is called the subject domain of the software. There may be additional elements added to the subject domain as a software development progresses such as user preferences, e-mail options or customers who are shopping online at a certain time. The elements of a subject domain depend on the selection of the environment that the problem exists which is determined by the selected software boundary. For example, when there exists a problem addressing a message broker software, the subject domain may comprise the received messages, queues, connected clients, etc.

The users and software communicate by assigning values to data attributes which map to properties of the elements of the subject-domain. The relational data model[74] is a value-based model that describe data in formal mathematical terms and create a consistent, logical representation of information. The relational model concepts allow organization of subject domain data in well-structured and homogeneous groups of attributes. Relational model is well-known in research and industry. However in the rest of this section we give a set of important concepts for the sake of completeness. The majority of the given terminology is adapted from [122] and [123].

The relational model makes use of a single structure to organize data: a variant of the mathematical concept of n-ary relation. A relation is a set of distinct n-tuples and each n-tuple is an ordered list of values, each having a domain. Relation can be represented in a natural way by means of a table in which every row corresponds to a tuple and every column contains values of an attribute of the relation (Fig 2.7). There are two levels of descriptions for a data model: the intentional level corresponds to a time invariant description of relations called as schemes and extensional level to the contents of the relations (instances) at a particular time.

Definition 1(Relation) A relation is set of ordered n-tuples of the form $< d_i, d_2, ..., d_n >$ such that each value dj is in the domain Dj, for $j = 1, 2, ..., n$. Alternatively, a symbolic name, called an attribute, is assigned to positions of domains in the sequence such that the relation can be expressed in terms of a set of attributes X. □

Definition 2(Tuple) A tuple over the set of attributes X is a function t that associates with each attribute $a_i \in X$ a value of the domain $dom(a_i)$. □

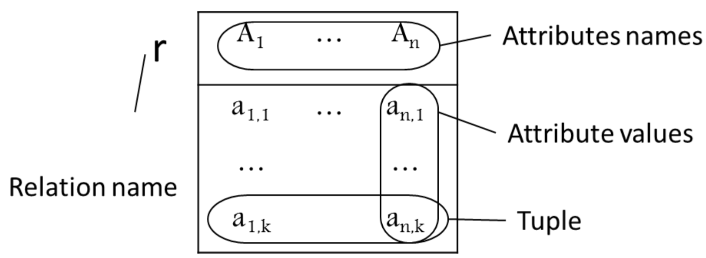


Figure 2.7: Relation

In a relation, all the tuples are distinct and thus no two tuples can have the same combination of values for all their attributes. There may be other sub-sets of attributes of a relation schema R with the property that no two tuples in any relation state r of R should have the same combination of values for these attributes.

Definition 3(Primary Key & Candidate Key & Super Key) A subset K of the attributes of a relation r is a superkey of r if it has the following property:

Unique identification: r does not contain two distinct tuples $t1$, $t2$ that agree on all the attributes in $K(t1,[K] = t2[K])$ A superkey that satisfies the following property is a candidate key:

Minimality: no proper subset of K possesses the unique identification property. There can be more than one candidate keys in a relation, a selected candidate key by a data model designer is called the Primary Key. □

Definition 4(Prime & Non-Prime Attribute) An attribute of relation schema R is called a prime attribute of R if it is a member of some candidate key of R. Otherwise it is called a non-prime attribute. □

Functional dependency is a formal constraint among attributes that is the main tool for formally measuring the appropriateness of groupings of attributes into relation schemes. Certain desirable properties of relational schemes have been defined in normal forms which are specified in terms of functional dependencies. The process of normalization consists of analyzing relations to meet increasingly more stringent normal forms leading to progressively better groupings of attributes.

Definition 5(Functional Dependency) Let R be a relation schema, and $X, Y \in R$ sets of attributes. The functional dependency $X \rightarrow Y$ holds on R if for every state of R, X determines Y such that if $t1[X] = t2[X]$, then $t1[Y] = t2[Y]$ in any relation instance. □

Definition 6(Trivial & Non-Trivial Functional Dependencies) A functional dependency $X \rightarrow Y$ is said to be trivial if $X \subseteq Y$ otherwise, it is nontrivial. □

Definition 7(Full & Partial Functional Dependency) A functional dependency $X \rightarrow Y$ is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any longer; that is, for any attribute $A \in X$, $(X - A)$ does not functionally determine Y. A functional dependency $X \rightarrow Y$ is partial if for some attribute $A \in X$, $(X - A) \rightarrow Y)$. □

Definition 8(1NF-First Normal Form) A relation scheme $R(X)$ is in first normal form, or flat, if every attribute in X is simple. Otherwise, it is nested. □

Definition 9(2NF-Second Normal Form) A relation schema R is in 2NF if every non-prime attribute $A \in R$ is fully functional dependent on the primary key of R. □

Definition 10(3NF-Third Normal Form) A relation schema R is in third normal form if, whenever a non-trivial functional dependency $X \rightarrow A$ holds in R, either(a) X is a superkey of R, or (b) A is a prime attribute of R. □

Relational algebra with a family of operators is a query language that is used to extract relations from the set of relations. The operators operate on one or more relation to yield another relation. The standard operators of relational algebra are:

1) Union ($\cup$): Two relations that belong to the same scheme can be combined into one using a standard set-theoretic union where duplicate tuples are eliminated. Two relations $R(A_1, \ldots, A_n)$ and $T(B_1, \ldots, B_n)$ are said to be union compatible if they have the same degree n and if $dom(A) = dom(B)$ for $1 \leq i \leq n$.

2) Difference ($-$): The expression $R - S$ describes a relation consisting of the set of tuples in R which are not in S.

4) Intersection ($\cap$): The expression $R \cap S$ describes a relation consisting of the set of tuples in R which are also in S .

3) Cartesian Product (X): RXT indicates the relation $Q(A_1, \ldots, A_n, B_1, \ldots, B_m)$ that is formed by combining tuples of relations $R(A_1, \ldots, A_n)$ and $T(B_1, \ldots, B_n)$ in a combinatorial fashion.

4) Projection (π): When applied to a set of tuples and given a list of attributes, the unary projection operation returns the same set of tuples, discarding other attributes of the relation. If any duplicates were created during this process, they are eliminated.

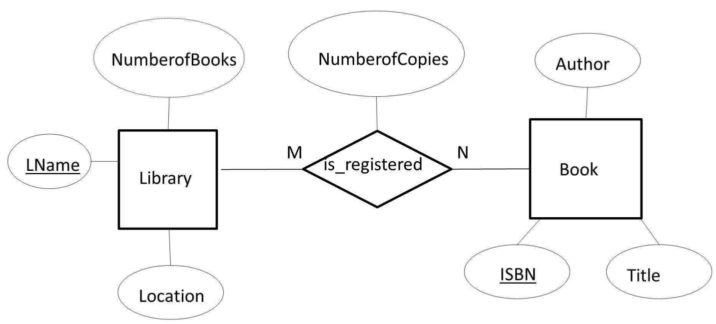
5) Selection (σ): Given a relation R, this unary operator applies a predicate (selection condition) to each tuple in the relation. If the tuple satisfies the predicate, it is added to the set of result tuples of the selection expression; otherwise, it is ignored.

Relational model defines normal forms based on the functional dependencies between attributes of the relations. When normalization is applied to a data model, redundancies of various types which can lead to anomalies and inconsistencies can be eliminated.

Initially, relational model was proposed for specifying the structures of data stored in software systems hence there was a semantic gap between the subject domain and the operational data models. Entity-Relationship based techniques [124] [122] were developed as a remedy and they have been the most widely used techniques where entity and relation concepts have become the terms for capturing important semantic information of the subject domain and the technique have provided a formal way to map conceptual schema down-to a logical relational model.

Example 2.1

Let us consider a library management system that is used to keep a track of books registered to several libraries of a university. Copies of books are distributed to libraries by the library managers. According to the specifications it is identified that a library software system needs to maintain data about books, libraries and the books registered for each library. The entity-relationship diagram and the resulting 3NF logical relation schema is given in Fig 2.8. Accordingly three relations, *BOOK*, *LIBRARY* and *LIBRARY_REGISTRY* are identified in the system and they are maintained as new books arrive and distributed to libraries. ■



(a)

BOOKS	ISBN	Title	Author
	1111128	Moby Dick	Melville
	4441229	Don Quixote	Cervantes
	2337841	Graphs	Berge
	7983234	Vehicles	Braitenberg
	9823473	COSMOS	Sagan

LIBRARY	Lname	Location
	PublicLib	Main Campus
	ScienceLib	Main Campus
	Kidslib	East Campus

LIBRARY_REGISTRY	ISBN	Lname	#Copies
	1111128	PublicLib	1
	1111128	KidsLib	3
	4441229	KidsLib	2
	2337841	ScienceLib	4

(b)

Figure 2.8: E-R Diagram and 3NF Relations for a Library Management System

A set of extensions to relational model was proposed, to fill the gap mainly with the motivation of representing complex data structures which was not possible in a first normal form scheme

[125][126]. First normal form of the relational model restricts values in a relation to scalars, thus tuples in a relation are flat.

Definition 11(Simple Attribute) Simple Attribute is an attribute of a relation whose domain contains only atomic values (scalars). □

Definition 12(Complex Attribute) A complex attribute is an either multivalued or relation-valued attribute of a relation. An attribute is multivalued if its possible values are sets of values;in this case the domain is the powerset of a given set of a scalar domain. An attribute is relation-valued if its possible values are tuples (of values); in this case, the domain is a relation. Complex attributes are set-valued. □

The nested relational model relaxes the first normal and hierarchical complex objects were modeled by recursively defining domains of attributes to be either scalar or set-valued where sets may comprise further relations. The notion of nested relation is the natural extension of the notion of a relation, e.g. it is a set of attributes that represent homogeneous tuples. Nested Relation structure can be graphically represented as trees [123] [127].

Definition 13(Nested Relation Scheme Tree) Nested Relation Scheme can be represented by means of a tree with three types of nodes that correspond to set (⊛) and tuple structure(⊗) constructors and atomic domains. Each atomic or set node has a label, which is the relation name for the root and the corresponding attribute name for each of the others. (1) the root is always a set node; (2) each set node has one and only one child, which is a tuple structure node; (3) the children of each tuple structure node are set nodes or leaves; and (4) each leaf is an atomic node and vice versa. □

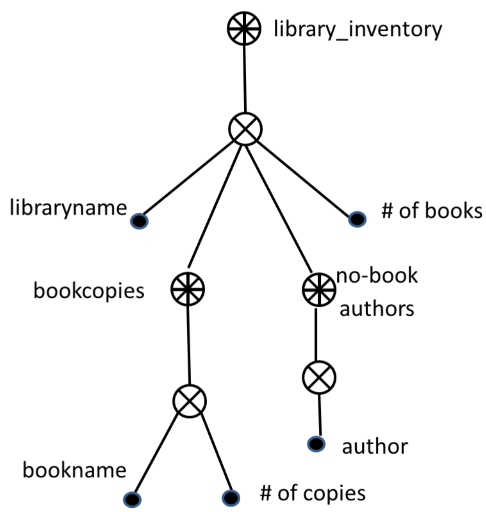
The construction of complex objects using set and tuple based attributes have been adopted in defining structural properties of the classes in object-oriented data modeling techniques and object oriented database development [128] [129] [130][131].

Example 2.2

Let us extend example 2.1 such that at the end of each year, the library managers takes a library inventory report from the system to be used for inventory counting for each library. The report includes information the library name, the total count of registered books, the author names of the books whose copies are not registered to the library and the count and name of registered book for each library. The report data is presented as a nested relation and a tree is shown in Fig 2.9. The relation LIBRARY _INVENTORY comprises attributes of $(LibraryName, Total\#ofBooks, BookCopies)$. Book Copies is a set valued complex attribute (relation) that comprises of tuples of type $< bookname, \#ofCopies >$ that belong to another relation *Book*. Authors are a set-valued complex attribute that comprises simple attributes. ■

LIBRARY_INVENTORY	Library Name	Total # of Books	No-book Authors	Book Copies	
				BookName	# of Copies
	KidsLib	5	{Berge, Sagan, Braitenberg}	Moby Dick	3
				Don Quixote	2
	SciLib	1	{Berge}	Graphs	1

(a) Nested Relation LIBRARY _INVENTORY



(b) Tree Representation

Figure 2.9: Nested Relation Example

CHAPTER 3

A MEASUREMENT MODEL FOR FSM

In this research our focus is on the functionality attribute of software.Having given a precise definition for functionality and addressing the specification as the software artifacts that embody functionality attribute (Section 2.2), in this chapter, we present a Software Functionality Model for Functional Size Measurement (4FSM). The primary motivation in the design of 4FSM is to improve consistency in specific model construction phase of a measurement process (step-2,Section 1.1.2) by providing a formal software meta-model (step-1, Section1.1.2). The identified sources of discrepancies in the exploratory case studies and the results of the literature review provides us with the inputs for setting the design targets for our model.

Section 3.1 gives the context for software functionality,the 4FSM constructs and rules and 4FSM model construction procedure. In Section 3.2 a mapping between COSMIC FSM method and 4FSM is given for demonstrating how a FSM method can be used with the 4FSM model.

3.1 Model Structure

3.1.1 The Context for the Model

Software comprises interacting components and serves a useful purpose as a whole for several entities that exist in the environment it operates (Fig 3.1). Software interacts with the environment via *functional users* which have *functional user roles* in the environment. The interaction between the software and the functional users is governed by stimulus-response (s-r) behavior. *Stimulus-response* (s-r) pairs constitute the chunks of software behavior. Software reaction upon the occurrence of an event private to the environment involves a single s-r pair or a sequence of s-r pairs controlled by its functional users. The desired effects of the software behavior in the environment are realized via the functional users of the software. Functional users interact the system by exchanging *functional messages* across a *boundary* in the course of s-r pairs. A functional message is either a *control command*or a *data group*. Data groups are organized according to *internal presentation* and *external presentation* styles. A s-r pair interacts with the functional users and the *Surrogate Data Model(SDM* component of the software. S-r pairs utilize *functional message capabilities* of the software in order to exhibit the specified behavior and produce the desired effects.

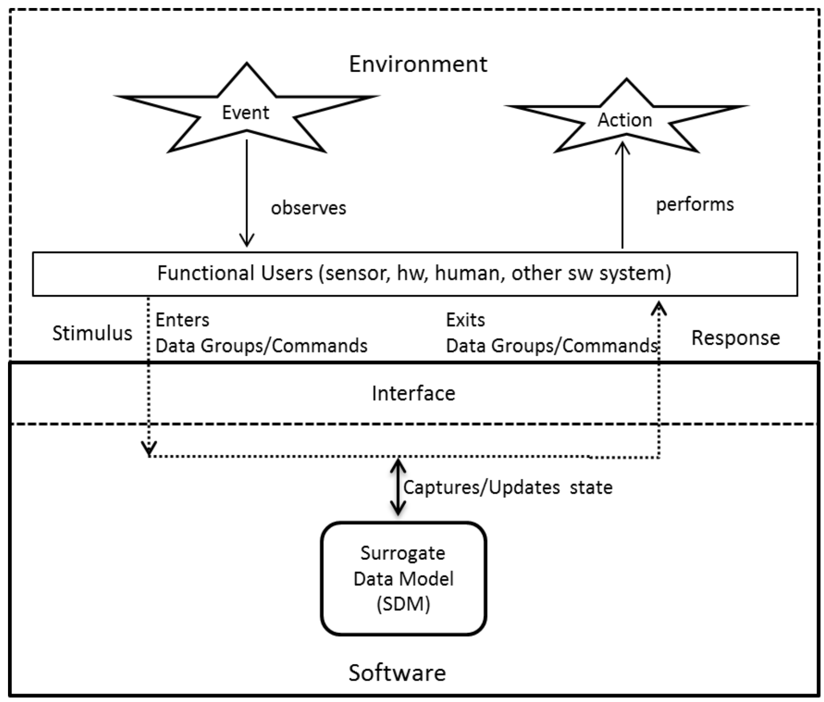


Figure 3.1: Software Context for 4FSM.

Definition 14(Functional User & Functional User Roles) A functional user is an external entity that interacts with the software by exchanging functional messages across interfaces such as GUIs, sensor interfaces or application programming interfaces (API). Functional users have assigned roles, which are distinguished with respect to the set of unique actions they perform in the environment. The set of roles are called Functional User Roles. The functional users and functional user roles are mapped by a relation T such that $T : FR \twoheadleftarrow\!\!\twoheadrightarrow FUR$, where $FU = \{fu_1 ... fu_n\}$ is the set of functional users and $FUR = \{fur_1, ..., fur_m\}$ is the set of functional user roles and $n, m \in \mathbb{I}$. □

Functional users are a part of the environment such as hardware devices (e.g. sensors, switches, actuators) and they contribute to the desired behavior of the environment.

Example 3.1

All people that purchase items from a shopping site are assigned ”customer” role and a subset of them who also administer the system are also assigned an ”administrator” role. In a traffic light control software, the green and red light software controlled switches have different functional user roles distinguished by their unique actions. When the first switch turns red light on the the drivers stop driving, when the other switch turns green light on the drivers start driving. ■

Definition 15(Boundary) The set of all interfaces through which software interacts with the functional users constitutes a conceptual *boundary* between the software and functional users. Functional users lie outside the software side of the boundary. □

3.1.2 4FSM Stimulus-Response Behavior

Definition 16(Stimulus) A Stimulus is a visible event (shared phenomena) occurs at the software interface and that is generated by a functional user. □

Stimulus are generated by a subset of the functional users of the software that observe the environment and interrupt the system as a consequence to events or conditions external to the software. While they can be in the form of requests, button clicks, sensor signals, they can also be timed or temporal events which are all included in the vocabulary of specifications. The set of stimuli statements in a specification is denoted as SS.

Definition 17(Response) A Response is the reaction of the software to the communicated stimulus. □

The notion of a response is to enable software to directly exert control over the environment enabling, enforcing, preventing events in the environment or implicitly affect it by providing information to the environment. Hence, they contribute the production of the desired effects of software in the environment in accordance with the specified behavior. The set of response statements in a specification is denoted as RS.

Definition 18(Stimulus-Response Pair) The stimulus and a response to the stimulus is called a stimulus-response (s-r) pair where the latter is causally dependent to the former. A formal definition for s-r pair is given in Definition30.

□

Example 3.2

Stimulus-response behavior can be observed in the following examples:

- Upon the request of a student to apply for a course registration, the registrant fills the students' details in a course application web form and submits it. The system stores the application details and returns a confirmation message.
- A smoke sensor detects smoke in the room where it is installed and stimulates the software in order to turn on the buzzer in security officers room.
- A digital display displays weather temperature and current time, alternating in ten second intervals.

In the above examples, the registrant, the sensor and timers are the functional users that observe the environment. The responses are recording the new applicant information, turning the buzzer on, displaying the temperature and displaying the clock time. The desired effects are having the student registered in the school, getting the security staff alerted that there is a fire and providing people with weather and time information so they can plan their actions accordingly. ■

The consequent software behavior upon reception of a stimulus can be different depending on the situation and condition at the instance when a stimulus is instantiated. In other words, the response to a stimulus is determined by state of the subject domain (environment) and the state of the software. For example, the registry stimulus may result in a warning message if the student has already registered or the course capacity is full.

The desired effects to the environment may be delivered by occurrence of a single s-r pair or through sequences of s-r pairs.

Example 3.3

When a professor requests the list of application for the course to prepare a class roster and software returns the applicants list, the desired effect is immediately realized in a single stimulus-response pair. On the other side, when a person at a floor requests an elevator by pushing the request button, the elevator control software immediately sends a command to start the lift-motor in the direction to the floor. The subsequent stimulus-response could be a sensor sensing the cage arriving to the floor and the controller sending a stop signal to the motor and an open signal to the door controller and so on. The desired effect is produced when the cage is ready at the requested floor via a sequence of s-r pairs. ■

While stimulus are visible events at the interface with the environment, software responses enable the satisfaction of desired effects by producing events and state changes which are visible in the environment. Sending a lights on signal to a car-bulb switch constitutes an example for the former where storing a new customer's contact details in a CRM software is an example for the latter. The software state change is not an end itself. However, the state change is the mechanism that implicitly maintains an effect distinguished by functional users in a form to be delivered via subsequent s-r pairs. When a request for list of all customers is received after a state change, the response will output the new customer's details in the list.

3.1.3 Representing Data Concepts in 4FSM

4FSM defines two relational modeling styles for logical presentation of data called internal and external presentation. First is used to organize data maintained and persisted inside the software boundary and the second for the data exchanged with the functional users across the boundary. Both presentations are at the intentional level such that subject domain attributes are organized as relational schemes. Both presentations are logical thus they are independent from physical data storage, presentation and collection structures and physical mediums such as binary files, xml files, graphs, etc.

Internal Presentation of Subject Domain Data Being the central conceptual component in the analysis, specification and design of systems such as management information systems or business applications, a model of the subject domain is maintained by the software and kept in a more or less synchronized correspondence with the subject domain of the software. The data model acts as a surrogate of the subject-domain. The notion of the model is providing information to the software that can not be conveniently obtained from the subject domain itself when it is needed. We name the data model of the subject-domain as surrogate data model (SDM) in our measurement model.

Definition 19(Surrogate Data Model(SDM)) SDM is a logical and semantic model of persistent data obtained from conceptual models. It is not a physical component of a solution system such as a database instance operated by a RDBMS nor does it have any sort of programmed behavior. □

SDM simply represents a structure of subject domain data and a state that can be captured or updated as per the specified response behavior. SDM structure maintains flat base relations in 3NF of the relational model. SDM comprises a set of finite relations $R_s = \{r_1, r_2, ..., r_n\}$ where the relation $r_j \in R_s$ is composed of a set of simple attributes $A_j = \{a_j1, a_j2, ..., a_jn\}$. SDM allows constant relations, the relations whose tuples have constant values and that do not belong to the SDM state, for the reason that they provide data that enable an appropriate software response (i.e., display output, join other relations, perform calculations). State updates are visible (distinguished) at the interface. State updates are not subject to integrity constraints and null values are allowed in any resulting state change. SDM interacts only with software in the course of response generation.

External Presentation of Subject Domain Data The mechanism for the exchange of subject domain information with the functional users is the assignment of values to data attributes that describes an aspect of concepts and realities in functional users' world. The sets of data attributes that are being assigned values and exchanged across the boundary are associated to relations that model the data of objects in the subject domain. The data structure of the objects are defined in relations such that they are in a correct form only if the functional user interpretation is expressed by the object interpretations.

Conceptual data modeling techniques are mostly used to model data to be stored in the system, reasoning that the values for data attributes of classes that are being communicated but not stored can be derived or calculated from stored data and further complexity and redundancy is avoided. However, such classes have possible instances, each with a distinguishing characteristic and they qualify as individual entities. The classes are instantiated as individual objects while software receives information from the functional users and outputs information to the functional users. The values for data attributes of such classes map to the values of the attributes from

- the persistent relations of the SDM,
- the attributes that are specified as functions of SDM relations,
- the derived relations obtained by applying relational algebra expressions to the relations of the SDM or
- the transient relations that represent an element of the subject-domain but not maintained in SDM.

The relations may have a simple structure where all its attributes are scalars or they may have a hierarchical composition such that the attributes may have a complex structure. Relations are presented as flat or nested relations. The relations are not subject to SDM rules such that they are not necessarily flat and in 3NF. Thus, the subject domain related information available at the interface is a set of finite relations, $R_t = \{r_1, r_2, ..., r_t\}$ and the relation $r_t \in R_t$ is composed of a set of attributes either simple or complex $A_t = \{a_t1, a_t2, ..., a_tn\}$. R_t is not necessarily disjoint from the relations of SDM (R_s). We assume single attribute relations such as errors and confirmation messages. Subject domain related information that software exchanges with the functional users and SDM are expressed in terms of R_t and R_s, respectively.

Definition 20(Data Element) A Data Element is the smallest piece of information that can not be subdivided into meaningful pieces. Data elements have scalar domains. □

There are two types of data elements the functional users and the software exchange. First type refers to the simple data attributes that describe the properties and states of the elements of the subject-domain and the second type are command parameters that enables software to control actions in the environment and functional users to control the software behavior. Data Groups and Control Commands are the higher level data constructs that are formed by simple or complex data attributes and command parameters respectively.

Definition 21(Data Group) Data groups are the logically related and cohesive groups of data attributes of a single relation. Data groups are the members of the flat and nested relations subject domain data organized accordingly to internal and external presentation. Thus, a datagroup is a relation $r, r \in R_t \cup R_s$. A Data group comprise simple or complex data attributes. □

By definition, nested relations contain subsets of homogeneous data attributes which may be relations themselves. This implies that a nested relation at a minimum includes a complex attribute. A top-down decomposition is applied in order to identify further homogeneous groups of data contained in a nested relation. For each complex attribute in a nested relation scheme, a separate data group is identified. This identification process is iterated recursively from top level nested relation down to all simple attributes and every complex attribute is identified as a separate data group.

Example 3.4

Let us consider the library example 2.2 in section 2.2.3. As shown in the tree representation (Fig 2.8), the top level node is the libraryinventory relation and by definition a relation is a complex attribute since it is set-valued. The no-book authors and bookcopies are the other set-valued (nodes indicated as (⊛)) complex attributes of the relation that is nested into the top level relation. Therefore, three data groups are identified from the tree representation. ■

The environment controlled events stimulate the software and software controlled events are realized in the environment by passing control commands to software and functional users respectively.

Definition 22(Control Command) A Control Command initiates a single action of the functional users or the software when received. A control command that is sent from the environment (environment controlled) and sent from the software (software controlled) is denoted as c_e and c_s respectively. The set of all control commands is $CC = (CE \cup CS)$ where $CE = \{ce_1, ..., ce_i\}$ and $CS = \{cs_1, ..., cs_j\}$ $i, j \in \mathbb{I}$. A Control Command,$cc_k \in CC$, comprise a set of cohesive parameters $P_k = \{p_k 1, p_k 2, ..., p_k n\}$ $k, n \in \mathbb{I}$. □

Example 3.5

Consider a query publications functionality in a publishers' online site (Fig 2.5). The command is ”query” and one of its parameters is sorting order that takes values ascending or descending. The other parameter is search scope(meta data/full text) and so on. ■

The data groups and the control commands constitute the words of the shared vocabulary between the functional users and the software. In our model we abstract the groups of both types of data to the concept of Functional Message.

Definition 23(Functional Message) Functional message is a cohesive group of data elements. A functional message represents both data groups and control commands, i.e. $fm \in (CC \cup R_t \cup R_s)$. The set of functional messages is denoted as FM. □

Although a specification may require that in the course of s-r pairs, different subsets of the data elements in a data group or in a command are exchanged, they are considered to be instances of the same fm since they belong to the same relation or the same command. Thus, $\mathcal{P}_{\geq 1}(A_j)$ and $\mathcal{P}_{\geq 1}(P_k)$ always maps to relation r_j and command cc_k, respectively.

3.1.4 4FSM Model Constructs and Rules

Definition 24(Functional Message Capability) A Functional Message Capability (FMC) is a software capability that enable directed passage of a single functional message (datagroup or command) from senders to the recipients. Senders and recipients are selected from the set of functional user roles, SDM and the software. FMCs enable s-r pairs to exhibit the specified behavior such that

- functional users can stimulate the software
- the response can collect information about the environment via functional users, or from SDM
- the response can generate the desired effects via directly passing control or subject-domain information to the functional users,
- the response can update SDM state such that they are distinguished at the interface

□

The effect of a FMC utilization in a s-r pair is visible at the software interface by functional users of the software upon the completion of a response. They can be observed instantaneously during the course of a stimulus-response pair where the software controls the functional users or provides them with information about the subject domain or the effects can be observed later in the course of another stimulus-response pair after a state change in SDM. The states changes on SDM leads to SDM states that are visible by being distinguished at the interface.

There are four categories of FMCs:

Definition 25(Enter FMC) An Enter FMC enable passage of stimuli and collection of information about the environment across the boundary via functional users. An Enter FMC is denoted as tuple E $\langle rec, fm \rangle$ where $rec \in FUR$ and $fm \in (CEUR_t)$. The set of all Enter FMCs is EnterCapabilities. □

Definition 26(Exit FMC) An Exit FMC enable the passage of control commands and provide information to the environment across the boundary by the software via functional users. An Exit FMC is denoted as tuple $X \langle rec, fm \rangle$ where $rec \in FUR$ and $fm \in CSUR_t$. The set of all Exit FMCs is ExitCapabilities. □

Definition 27(Update FMC) An Update FMC enables the passage of a data group from software to the SDM and update of SDM state by modifying the relation data group is associated. An Update FMC is denoted as tuple $U \langle rec, fm \rangle$ where $rec \in \{SDM\}$ and $fm \in R_s$. The set of all Update FMCs is UpdateCapabilities. □

A response can update the state of SDM utilizing an Update FMC in accordance with Rule-1.

RULE-1: A relation can be updated by adding, deleting the entire tuple in a relation or updating the values of a subset of its attributes. In order to update and delete the subset ST of tuples of a relation R ,$ST \subset R$ the response part should have collected sufficient information from the environment or SDM such that it should be able to uniquely select the tuples to update or delete. This requires that the response part should have the candidate key (primary or unique key) before updating or deleting tuples. If all tuples of relation R, are updated or deleted then it is not required.

Definition 28(Capture FMC) A Capture FMC enable the passage of a data group from software to the SDM and capturing the state of a single relation from the SDM state. AN Capture FMC is denoted as tuples $C \langle rec, fm \rangle$ where $rec \in \{software\}$ and $fm \in R_s$. The set of all Capture FMCs is CaptureCapabilities. □

Definition 29(FMC Type) The type of a FMC is a sub-class of a category of FMCs such that it belongs to a FMC category and it has the same rec and fm components. □

Stimulus-Response pairs are the homogeneous chunks of specified software behavior that is initiated by stimulation of software by a functional user and is responded by the software performing the specified response and contributing to desired events and states visible at the interface via a set of FMCs. A stimulus is always realized in the form of a Enter FMC that conveys an environment controlled control command cc, $cc \in CE$ recognized by the software. All FMC types utilized in a response constitute the capability of a response to collect information from the environment and produce visible effects in the environment.

Definition 30(Stimulus-Response Pair) A stimulus-response pair is a quadruple $\langle s, r, fs, fr \rangle$ where: $MF \in$

- s is a string stimulus statement, $s \in SS$
- r is a string response statement, $r \in RS$
- *fs*, is a FMC needed to communicate a stimulus, i.e $fs \in CM$ where $CM = (cm \in EnterCapabilities | cm[fm] \in CE)$
- *fr*, is a set of functional FMCs that the response needs to utilize to exhibit the specified behavior, i.e.
  $fr \in \mathcal{P}(T)$ where $T = (rm \in FMCS | rm[fm] \notin CE)$ and
  $FMCS = (EnterCapabilities \cup ExitCapabilities \cup UpdateCapabilities \cup$
  $CaptureCapabilities)$

The set of all s-r tuples identified from specifications is called a 4FSM stimulus-response scheme, $SR = \{sr_1, ..., sr_t\}, t \in \mathbb{I}$. □

RULE-2: A stimulus occur as a consequence to external events that are private to the environment. They happen without the control of the software, hence they ”interrupt” the software. While the software is performing a response to a stimulus, the specification may require the interruption of another functional user external to the software via an Exit FMC and wait for one or more functional messages from the functional user in order to complete the response properly. The behavior of the functional user upon the reception of such stimulus is equivalent to the response behavior of the software. The response of the functional user may involve a FMC directed to the software. In such cases the reception of the functional message should not be considered to interrupt the software and lead to identification of another stimulus-response pair. The event controlled by the response of the functional user at the interface is anticipated by the software and is reasoned by the initial stimulation (interrupt) of the functional user by the software. Hence, it is not considered to interrupt the software. In summary, an events that occurs at the interface and generated by a functional user are identified as stimulus when the stimulus when they are not controlled by the software. The effects delivered by the functional users as a response to the stimulus generated by the software are attributed to the response part of the initial stimulus received by the software. This kind of interaction is usually observed in software-to-software interactions or in software responses that prompt users for additional information about the environment while the response generation has already been started. An example case for Rule-2 is illustrated in Figure 3.2

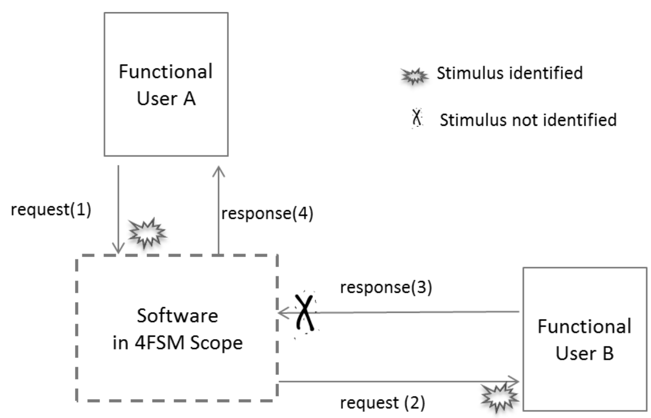


Figure 3.2: Illustration for Stimulus Identification Rule-2

RULE-3: The response part of a s-r pair may result in an effect such that resulting software behavior involves only an observable software state change in the environment but can not be expressed in terms of FMCs. Such responses are typical to complex user interfaces where state changes involve the mechanics of the interface such as navigational properties of a complex graphical user interface. In such cases fr component of the s-r tuple is ∅.

The FMCs that a s-r utilizes in fs and fr components are identified by identifying of all possible data elements (datagroups and commands) input from the environment, output to the environment, persisted in SDM and captured from SDM in the course of a s-r pair and establishing relations between them. Specifications should include sufficient detail for the identification of FMCs.

Example 3.6

Consider the following specification excerpt from an online order software specification: *"Customer cancels an order by requesting "cancel" from order main page. Software takes the selected ordernumber (orderno) from the interface deletes all related information for the order and returns all order numbers.*" Two relations of the SDM that represent order related information is given in Figure 3.3.

ORDER	OrderNo	CustomerNO	ShipmentAddress	Status
	B12098	12	Inf.Inst. METU-Ankara	Open

ORDER_ITEM	ProductCode	Quantity	OrderNo
	1088934	5	B12098

Figure 3.3: Orders and OrderItems

The functional user role, customer, stimulates the "cancel order" s-r pair via Enter FMC $\langle "customer", "requestcancel" \rangle$. Software will receive the details of the order to be deleted in the response via Enter FMC $\langle "customer", "ORDER" \rangle$. Corresponding Order tuple from ORDER relation and tuples from ORDER_ITEMS can be deleted via Update FMCs $\langle SDM, "ORDER" \rangle$ and $\langle SDM, "ORDER\_ITEM" \rangle$ respectively. Specifications imply that only a subset of relations in ORDER and ORDER_ITEM are to be deleted. Rule-1 given in FMC definition requires that intended tuples should be uniquely selected. Rule-1 is satisfied for ORDER since response has collected OrderNo information from environment and it is sufficient to select the tuple in ORDER. However, the response do not have the key (OrderNo, ProductNo) Rule-1 is not satisfied for ORDER_ITEM. Therefore further information for selection of the order items to be deleted from the SDM should be taken via Capture FMC $\langle SDM, "ORDER\_ITEM" \rangle$. Finally all tuples in ORDER relation should be retrieved via Capture FMC $\langle SDM, "ORDER" \rangle$ and displayed via Exit FMC $\langle SDM, "ORDER" \rangle$. A total of seven FMCs should be utilized to satisfy the specified behavior. ■

4FSM Scheme Normalization

In a 4FSM stimulus-response scheme, SR, the s-r pairs are distinguished into three disjoint sub-sets $ISR, OSR, NOSR$, i.e., $SR = ISR \cup OSR \cup NOSR$. Subset ISR refers to s-r pairs whose response can not be expressed in terms of FMCs. Formally, $ISR = (sr \in SR \mid sr[fr] = \emptyset)$. S-r pairs that pertain purely to interface mechanics (see RULE-3) belong to this subset.

Subset OSR contains stimuli-response pairs whose response, at a minimum, include either a FMC that is directed to a functional user or to the SDM. A s-r pair that satisfy this condition is called an *observable* s-r pair. Formally, the following predicate holds true for an observable s-r pair:

$isObservable(s-r) = \exists fmc : s-r[fr] \bullet fmc \in (ExitCapabilities \cup UpdateCapabilities)$

Subset $NOSR$ contains s-r pairs whose response is limited to collection of information from the environment or SDM. A member of set $NOSR$ is called a non-observable s-r pair. A *non-observable* s-r pair satisfies the following :

$isNonObservable(s-r) = \neg isObservable(s-r) \wedge$

$(\forall fmc : s-r[fr] \bullet fmc \in (EnterCapabilities \cup CaptureCapabilities))$.

In a s-r schema, one or more observable s-r pairs depend on non-observable s-r pairs for the collection of sufficient information from the environment and non-observable s-r pairs' response part become observable when their associated observable s-r pairs' responses are completed. Hence, there exists a relation $O : OSR \leftarrow\!\!\twoheadrightarrow NOSR$.

Example 3.7

For example, a user enters only the information about the environment executing s-r pairs sr1, sr2 (e.g. entering some information on a two-step wizard form) and saving the collected information in sr3. The dependency relation is $\{(sr3 \leftarrow\!\!\twoheadrightarrow sr1), (sr3 \leftarrow\!\!\twoheadrightarrow sr2)\}$. ■

Before the use of a 4FSM model in the final step (step-3 in Section 1.1.2)of a measurement process, given the set of all s-r pairs,SR, the following normalization steps should be followed in order to obtain homogeneous s-r schema:

1-All interface s-r pairs (ISR) are eliminated from the scheme. The resulting scheme is $SR' =$

$SR \setminus ISR$

2-If $NOSR \neq \emptyset$ a refinement is performed iteratively on all non-observable s-r pairs such that the response part (fr) of each non-observable pair is added to the response part of observable pairs that depend on them. The procedure is given in Figure 3.4 in Java style. The resulting scheme is $SR'' = SR' \setminus ISR$.

The scheme SR'' is called an *observable schema* and all members of the scheme are observable. A 4FSM model with an observable schema is said to be in the *measurable* form. In accordance with ISO 14143, each FMC utilization in an observable s-r pair of an observable 4FSM s-r schema is a Base Functional Component (BFC) of the 4FSM model.

```
Types:
class SR{String stimulus; String response ; Smapf fs; Rmapf fr;  }
class Smapf{ String stimulus; FMM sFMM; // stimulus mapping  function
class Rmapf{  String stimulus;   Set<FMM> rFMMs;  }//response mapping function
class FMM{ String type; String fm; String rec; }

Variables:
srset  // set of identified  SR
osr//observable SR
nosr //set of non-observable SR
measurableSR //set of measurable  SR

Initialize :
Set<SR> srset= set of identified SR pairs;

Procedure:
Iterator<SR> srit=  srset.iterator();
    while (srit.hasNext()) {
          SR sr= srit.next();
          Set flows= sr.fr.rFMMs;//return set of FMM that enable response
            if(flows.hasMembersofCategory("EnterMovement") || flows.hasMembersofCategory("ExitMovement")) {
                          osr.add(sr) ;
            }
    }
    nosr= srset.minus(osr);
    Iterator<SR> nobit=  nosr.iterator();
    while (nobit.hasNext()) {
                        SR dependentSR= nobit.next();
                        Iterator observableSR= osr.iterator();
                        while (observableSR.hasNext()){
                            SR observable=observableSR.next();
                                 if(dependentSR.depends(observable)){
                                     observable.fr.rFMMs.add(dependentSR.fr.rFMMs);
                                 }
                        }
     }

    Set measurableSR= osr; //measurable SR.

Output:
measurableSR //set of measurable SR
```

Figure 3.4: Procedure for an observable 4FSM schema

3.1.5 4FSM Model Construction Procedure for FSM

In order to construct a software functional model in accordance with the construct definitions, a series of activities should be performed in order to end-up with a 4FSM meta-model instance.

- Step-1: Identify Measurement Scope, Functional Users, Boundary: This step involves the selection of the measurand with respect to the measurement purpose. Measurement scope is a piece of software which can be distinguished from an environment via a boundary. It can be selected as the whole software of a system or it can be a software component that is a part of a software component composition. In order to be a valid selection, the specification artifacts that describes the external behavior of the software component should be available or derived. The specification must satisfy requirements in the environment and allow construction of a subject-domain that involves all types of different interfaces. When determination of the scope implies that all functional users (interacting entities), functional user roles and the boundary are identified. There must be an internal behavior or a design freedom inside the boundary that should be excluded from the scope.

- Step-2: Construct the Model: Once the boundary is determined, the model constructs are identified from the specifications. That is, SDM is established, functional messages, stimuli-response pairs and FMCs utilized in the course of each s-r are identified.

- Step-3: Apply the Normalization Procedure: A further processing is done on the set of identified stimuli-response pairs. Non-observable s-r which do not provide any information to the environment and control any action in the environment are eliminated while maintaining their contributions to software functionality by merging their FMCs to observable pairs of s-r. The procedure ends in a set of s-r pairs that are all observable.

3.2 Mapping to COSMIC Software Model

COSMIC FSM [54] is an ISO 14143 compliant model that attracts a growing interest from international community. The motivations for COSMIC have been not only to improve FPA concepts but extend the applicability of FSM to real-time software domain and resolve problems with respect to measurement theory. The method also have incorporated concepts from MARK II FPA and has been called a new generation FSM method[7] (Fig 3.5). In this section we compare the constructs of COSMIC FSM and 4FSM functionality models and summarize the corresponding concepts.

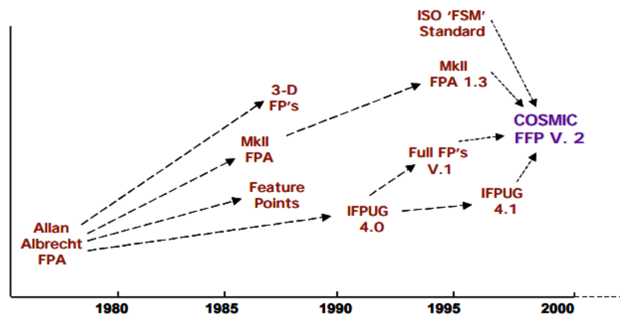


Figure 3.5: Evolution in FSM and COSMIC FSM Method-adapted from [7]

Scope, Boundary, Functional Users. The determination of the measurement scope concerns the selection of the piece of software which is the subject (measurand) of a measurement. The precise determination of scope is given in terms of Boundary, Functional Users in both COSMIC and 4FSM. COSMIC defines boundary as a "conceptual interface between the software being measured and its functional users." This definition is consistent with 4FSM definition of boundary where it is the set of interfaces between the software and functional users. Similarly, the definition for Functional Users is defined as "A (type of) user that is a sender and/or an intended recipient of data" which corresponds to the concept of "functional user roles" in 4FSM. One subtle difference between the COSMIC FSM and 4FSM lies in the approach to functional user requirements. According to COSMIC, functional user requirements describe software in terms of functions that further broken-down and can be allocated to hardware, software or a combination of the two. The allocation is performed with respect to a physical computer architecture where software is organized into "layers" of the architecture (Fig3.6). COSMIC defines each component interacting with others in the same physical layer as a "peer component". COSMIC requires that the piece of software being measured should confine in a single layer. On the other hand, 4FSM views functional requirements as a specification of a single software component or one of its sub-components which describes the desired external behavior of software in terms of stimuli-response behavior that is expressed in terms of communication at the interface and states of a surrogate data model. 4FSM model construction is constrained with the interface description and should not involve any further decisions and assumptions about the internal structure and behavior of the software. It does not require associations on how the desired behavior emerges from such functions obtained by a top-down

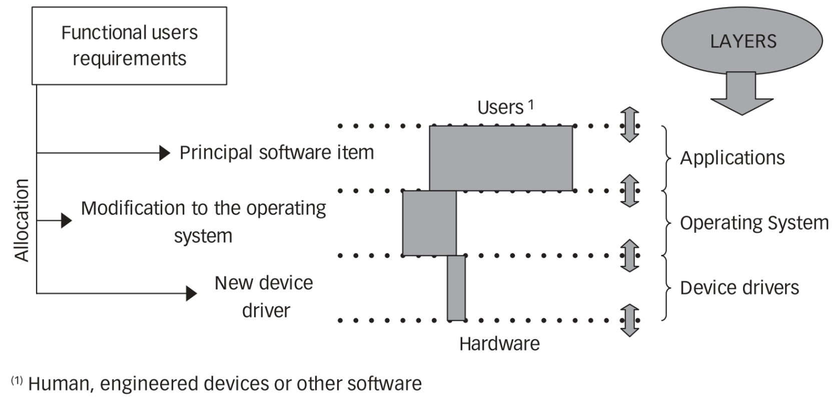


Figure 3.6: Allocation of FURS into Layers in COSMIC-adapted from [1]

breakdown. However the specifications draws the boundary of software by describing the interactions and the observable behavior at each interface. Therefore, the functionality of any piece of software at any layer can be modeled provided that it describes an interface between the software and an environment. 4FSM requires that there must exist a defined environment and a specification for the software being measured. However, COSMIC requires that the software piece to be measured should confine in a single layer, thus distinguishes the subject domains between the software components in different layers and constraints the measurement to be performed to a subject domain that pertains to a well-defined environment and that is shared between peers components and their functional users. In summary, there are no equivalent concepts of layers, peers in 4FSM , however, they both models include concerns for identification of a well-defined environment for a software component.

Functional Process. COSMIC defines a functional process as an elementary component of a set of Functional User Requirements comprising a unique, cohesive and independently executable set of data movements. It is triggered by a data movement (an Entry) from a functional user that informs the piece of software that the functional user has identified a triggering event. It is complete when it has executed all that is required to be done in response to the triggering event. A triggering event is defined as an event (something that happens) that causes a functional user of the piece of software to initiate (trigger) one or more functional processes. In a set of Functional User Requirements, each event which causes a functional user to trigger a functional process cannot be sub-divided for that set of FUR, and has either happened or it has not happened. As clear from the definitions, both COSMIC and 4FSM models agree

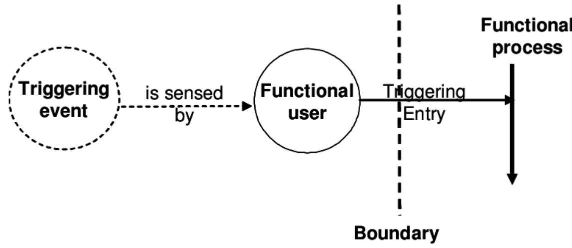


Figure 3.7: Relation between External Event and Functional Process-adapted from [8]

on the reactive view of software behavior and functional process roughly corresponds to the s-r pairs of the 4FSM. However, in COSMIC, the relations between the external events, triggering events and software behavior allows ambiguity such that the relation between external events and the consequent software behavior is not straightforward (Fig.3.7). The relation can be affected by the state of the environment and the state of the software, thus, can be situational, dynamic and complex and is typically expressed as scenarios. From the perspective of the 4FSM model, while for some external events, the specification may permit one-to-one association between an external event and a stimuli, an event can also be responded by traces of s-r pairs. Furthermore, there can be different traces of s-r pairs for two separate occurrence of the same event. It is a specification task to define s-r pairs as the chunks of behavior such that the satisfaction of behavior in the environment after the occurrence of an external event should be possible by a specification process that produces stimuli-response pairs taking domain knowledge and requirements as inputs. Therefore, identification of functional processes is not without ambiguity and permits the identification of a functional process for each different scenario when only the definition is taken into account. When the rules for identification of functional processes and examples given in the COSMIC guides are explored [75], further information is found that the concepts of s-r pair and functional process converge. In the guides, many s-r pairs are given as examples of valid functional processes or s-r pairs are selected as candidate functional processes.

Object of Interest and Datagroups. COSMIC defines an Object of Interest as any thing that is identified from the point of view of the Functional User Requirements. It may be any physical thing, as well as any conceptual object or part of a conceptual object in the world of the functional user about which the software is required to process and/or store data. A data group is ”a distinct, non empty, non ordered and non redundant set of data attributes where each included data attribute describes a complementary aspect of the same object of interest."

Object of Interests clearly maps to elements of the subject domain where associated data is represented in external and internal representations. Data Groups map to the flat and nested relations in $R_t \cup R_s$ of a 4FSM model. While mapping the nested relations to data groups further refinement is required. For messages in the form of nested relations further decomposition is applied in order to identify subsets of homogeneous data. An attribute of a nested relation $r \in R_t$ at a minimum includes a complex attribute. Complex attributes do not have scalar domains such they comprise, a single-valued complex attribute or a set-valued attribute. For every complex attribute in a nested relation scheme a separate functional message is identified. This identification process is iterated recursively until all attributes are identified as simple scalars, thus, every homogeneous data group that is recognized as complex attribute is identified and mapped to a COSMIC data group. In COSMIC the concept of data groups represent both commands and properties of the elements of the subject domain. In 4FSM, commands and data groups are explicitly distinguished.

Data Movement. Data Movement is a base functional component which moves a single data group type. Total functional size is obtained by summing the number of data movement type occurrences in each functional process. There are four sub-types of data movement types: Entry, Exit, Read and Write. An Entry (E) moves a data group from a functional user across the boundary into the functional process. An Exit (X) moves a data group from a functional process across the boundary to the functional user. A Read (R) moves a data group from persistent storage within reach of the functional process which requires it. A Write (W) moves a data group lying inside a functional process to persistent storage.

According to COSMIC model a functional process comprises sub-process of two types: data movements and data manipulations. Sub-processes are identified from procedural descriptions of the software response to an external event from the view point of its user. On the other side, 4FSM FMC types are declarative constructs such they represent the functional capabilities of software to enable appropriate s-r pairs. However, in COSMIC data manipulations are considered to be represented by data movements and their contributions to the functionality are attributed to associated data movements assuming an average distribution to each data movement. Thus, data manipulations are not assigned a numeric size value and do not directly contribute to the size of software functionality. Furthermore, normally, COSMIC does not distinguish between sub-groups of data within a data group such that any combination of attributes that involve a Object of Interest are considered identical as long as they are subsets of the same data group. Following these COSMIC definitions, the concept of data movement and the procedural description converge to 4FSM FMC concept and the declarative descriptions respectively. In summary, the concept of data movement types maps to FMCs types utilized in each s-r pair. The instances of E, R, X and W data movement types map to the members of $EnterCapabilities$, $ExitCapabilities$, $UpdateCapabilities$ and $CaptureCapabilities$, respectively.

In COSMIC, total functional size is calculated by summing the count of E,R,X,W data movements in identified functional processes. Therefore, COSMIC functional size of a 4FSM model is obtained by adding all FMCs in fs and fr components of s-r pairs in an observable 4FSM scheme.

In COSMIC, one of the Entry data movements is assumed to trigger the functional process. This arbitrary Entry data movement is defined as a triggering entry (Fig.3.7). This particular data movement has a dual role such that it can move an explicit data group of an OOI, it can move a command or both a command and a datagroup. In a 4FSM, commands and data groups are explicitly distinguished. Besides, a stimulus is always a command passed from a functional user to the software, thus utilizes a separate FMC. Similarly, a functional process may exit a datagroup, a command or both in a single data movement. In COSMIC, the commands are interpreted as a triggering event that starts a functional process in a functional user (usually other software) in the same sense as the Rule-2 of 4FSM stimulus-response definition. Although this differentiation does not indicate a conceptual difference between 4FSM and COSMIC models, it needs to taken into account while comparing size values of a 4FSM and a COSMIC model. Since COSMIC size directly depends on the number of data movements and may lead to identification of extra data movements when a triggering entry or an exit conveys both a command and a data group. Table 3.1 gives a summary of COSMIC and 4FSM concept mappings.

Table 3.1: 4FSM and COSMIC Concept Mapping

COSMIC Concept	4FSM Concept
Functional User (Type)	Functional User (Role)
Boundary	Boundary
Functional Process	S-R pair in an observable scheme
Object of Interest	Any element of the Subject Domain
Data Group	Data Group or Command
Triggering Entry	Stimulus
Layer	-
Data Movement	Functional Message Capability

CHAPTER 4

CASE STUDIES

In this chapter, we first briefly discuss the case study as an empirical research strategy. Section 4.2 presents two exploratory case studies conducted in order to investigate sources of inconsistencies in functional size measurement results and discusses their results.

Section 4.3 presents two case studies conducted for validation of 4FSM. In the section, the results of the applications of 4FSM to the requirements specifications were analyzed and discussed.

4.1 Case Study in Research Design

Case study is an empirical research strategy commonly used in social and applied sciences[132] [133] [134] [135].They are used for investigating a phenomenon in a complex settings and within its real-life context for the purposes of understanding and explanation or constructing a theory [132] [136]. Case studies are generally observational or descriptive in nature, however they can be relational such that they are used to gain deep insights into chains of causes and effects. They can be based on qualitative as well as quantitative evidence. They can be applied both prospectively and retrospectively. Due to the versatility brought by such properties, case studies have also become popular in software engineering where they are used for understanding, describing and explaining problems, or demonstrating the adequacy and extend of software engineering methods and models, thus, for validation of research results [137].

The design of a case study can be basically characterized by two distinct properties[132]. First refers to the replication nature of the case study. When the design follow a replication logic it is a multiple-case and otherwise it is called a single-case study. Second distinction is made in the characteristics of its units of analysis. The design is called holistic if the units of analysis are identical; otherwise is called to have an embedded design.

In the two exploratory case studies and one validation study explained in this section multiple-case design were used. A multiple case often considered more compelling as it strengthens the results by replicating the pattern-matching and increasing confidence in the robustness of the theory. Since the contexts of multiple cases differ to some extent; the same conclusions derived from multiple studies have stronger external generalizability [132][137]. The retrospective validation study in Case 3 has a single-case due to the limited case availability. The design of the case studies were holistic such that a size measurement result is selected as the single unit of analysis in every repetition. The replication approach we followed in our multiple-case study design is depicted in Figure 4.1 (based on [132]).

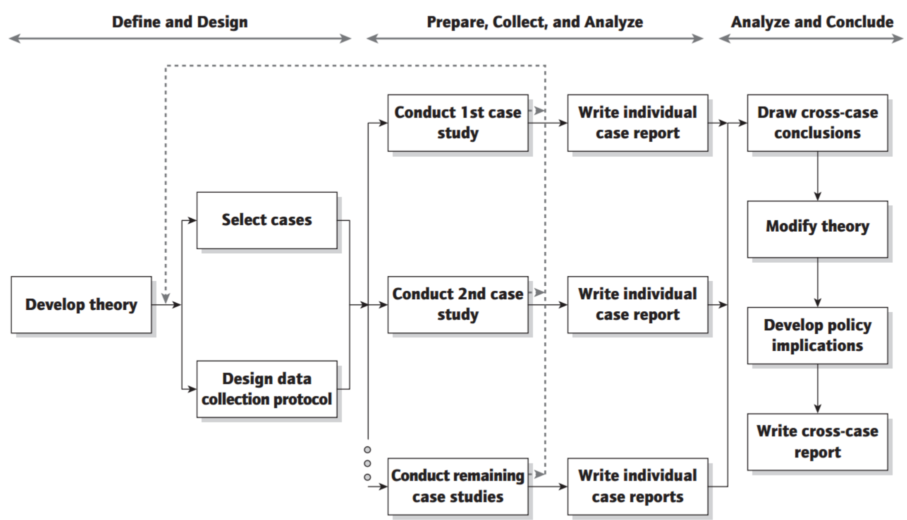


Figure 4.1: Case Study Method

4.2 Exploratory Case Studies on the Reliability of FSM Results

In our previous research studies and measurement experiences on FSM we observed inconsistencies between the measurement results for the same set of functional requirements [26][27]. When the results were analyzed subjective interpretations of the requirements and method rules explained a considerable portion of the differences in the measurement results. Besides, we observed a set of generic functional requirement properties that we found to relate to the poor results and creating challenges in measurements. These observations motivated us to study the reliability of measurement results. We particularly questioned the effect of the inconsistencies on the reliability of the results and software requirement patterns that constitute the root causes of discrepancies. In this section, we present two case studies designed for empirical evaluation of the measurement results obtained by applying the COSMIC method. The case studies were conducted by SMRG at Informatics Institute, METU as part of a research project [138].

Both studies were driven by the following research questions:

RQ1. What is the effect of measurement errors on the reliability of the measurement results?
RQ2. What are the common patterns of measurement errors?

In RQ1, our goal is to understand how measurement errors impact the reliability of the results and to what extend. In RQ2, we particularly look for associations between the errors such that the commonalities in errors can be identified, their sources can be identified and searched for patterns. The motivation was that when such information is available the problems that lead to inconsistent results can be well defined and distinguished. Furthermore, subsequent studies can be designed for improving measurement performances such that the opportunities can be explored for:

- improving method structure by suggesting precise meta-models and measurement procedures
- improving results by further method documentation and proper training and
- avoiding errors a-priori by proving feedback on patterns of errors to the FSM community

The main focus of the empirical research on the reliability of FSM methods has been the assessment of the ”reproducibility” characteristics of the measurement results where the closeness between the measurement results of different subjects using the same method have been investigated [20] [21] [23] [22] [29]. The data used in the studies were single numeric values obtained from measurers. Their approach did not include assessment of the models constructed in accordance with FSM method meta-models.

However, the answers to our research questions requires the investigation of the causal relations between requirement specifications and the functionality models constructed according to the selected FSM method meta-model. Furthermore, it necessitates data on the interrelations established between model components. Therefore, we extend the measurement result assessment approach to the constructed models and the data we collect are not limited to the calculated numeric size but also include details on the constructed models. Therefore, the measurement results being analyzed should be understood as the recorded presentation of the models constructed by following the meta-model definitions and associated rules of the FSM method.

In the case studies, an error is defined as any result that is considered invalid as per the rules and definitions given in COSMIC FSM [8][75] or any result that can not be properly and adequately verified as per the rules and definitions given therein. Thus, the inconsistencies in results were identified in error units. Identification of errors requires a correct model of software functionality that has been constructed according to the rules and definitions of the selected method meta-model. However, there is not any other method to construct the correct software model for a COSMIC software model and obtain the ”true value” of functional size. One suggested and used way for obtaining a correct model of software functionality and its true value is based on the consensus of method experts [73][60]. In both studies the software models for analysis were constructed with the consensus of three COSMIC certified experts and a member of the COSMIC Advisory Board.

In both studies, a multiple-case design was followed in order to observe the measurement errors in the contexts of different measurers, products and discuss their reproducibility in different circumstances. The analysis were also performed by the experts. For our analysis purpose, we took measurement results not as a single number in functional size units but as a measurement report where all functional components identified by the measurer are recorded.

4.2.1 Data Analysis Procedure

The analysis of the measurement results in both studies was performed in two steps. In the first step, the functional components identified and reported by the measurers were compared to the keys that contain all valid COSMIC functional components in the requirement sets from the COSMIC experts viewpoint. The measurement report includes information on Functional Processes (FPs), Data Groups (DGs), Objects of Interests (OOIs) that DGs belong to and the Data Movements (DMs) each FP contains. When a FP matched a FP in the key they were marked as correct, when there is no corresponding FP in the key it was marked an incorrect. When a FP in the key was not detected and not recorded in the results it was marked as missing. Similar convention was followed for DMs; however, in order to be marked as correct, in addition to having a matching type (Enter, Read, Write, Exit) and DG, they were required to have the correct identification of the associated OOI of the DG being moved.

In the second step of the analysis, the requirements and errors were investigated for cause-effect relationships and the commonalities between the errors were explored. To support the inspection process a checklist was prepared by gathering several findings from our previous research and industrial experiences [26][27]and guides [8][75] (Table 4.1).

The data movements in an incorrect or a missing FP were marked as incorrect or missing. Whenever the analysis revealed that a reported FP is a combination of several correct FPs, then it was assumed to match the FP having the highest number of DMs and others were considered incorrect. The analysis results regarding the second step were presented in FP and DM based categories where the instances of measurement errors were associated to incorrect identification or subjective understanding of these measurement concepts.

Table 4.1: Potential Error Patterns

Parameter tables are considered as OOIs
Different Error Messages are considered as separate Exits
Attributes are considered as Data Groups
List Before Update/Delete is ignored
Retrieve Before Update/Delete is ignored
List is defined as a part of update/delete FP
Retrieve is defined as a part of another FP
Transient data group concept is ignored
Multi Pages are considered as separate FPs
Cascading Delete is ignored

We defined two indicators, Accuracy Rate (AR) and Component Detection Rate (CDR), in order to analyze the measurement results, measurement performances and the effect of errors on the results. AR and CDR are used for pairwise comparisons between individual measurements, measurers and groups of measurements. AR is the rate of correctly identified items to the sum of all correct, missing and incorrect items. CDR indicates the degree of success in identification of the measurement items; hence the rate of correct items to the items that should be identified in the requirements.

AR_x = # of Correctly Identified x/ # of (Incorrect + Missing+ Correctly Identified x)
CDR_x = # of Correctly Identified x / # of all valid x in the requirements identified by experts
where $x \in DM, FP, DMandFP$

4.2.2 Case 1: A Multiple-case study on Industrial Projects

4.2.2.1 Data Sources

The measurement results of twelve products developed in eight industrial software projects in three different organizations were analyzed (Appendix B). The measurements were performed by five measurers. The functional specifications of products were obtained from various software requirement artifacts. The software products to be measured were selected from the organizations where the measurers work as software engineers or project managers (Table 4.2). The measurers were graduate students that had taken a software project manage-

Table 4.2: Case Organizations Profiles

	Organization 1					Organization 2			Organization 3			
Quality System	CMMI[a] level 3					ISO/IEC 15504[b], CMMI level 3			CMMI level 3			
Software Product Name	AN	BN	MN	KA	GH	CN	SN	TN	TM	HB	TCL	DVTCL
Product Business Domain	Utilities	Finance	Finance	Human Resources	Human Resources	Change Management	Acquisition	Accounting, Acquisition, HR	Taxation	Taxation	Taxation	Taxation

[a] Capability Maturity Model Integration,[b]Software Process Improvement and Capability Determination

ment course as a part of their graduate programme. The course included six-hour COSMIC functional size measurement v3.0 training and the students had measured several projects as part of the course. The students passed the course with a grade of BB (75/100) or higher. The measures had a similar level of knowledge and experience in FSM and conducted the measurements as a part of project course which was a prerequisite for their graduation.

4.2.2.2 Conduct of the Case Study

Since the students perform the measurements in the conduct of their term project course, the measurement process spanned several weeks. They reported the measurement results on a spreadsheet template (Appendix A) that enables the recording of detailed measurement results. After the measurements, the experts analyzed the available requirement documents and measurements and identified and classified the errors. Interviews were performed with some of the measurers to obtain more information on the requirements, assumptions and the causes of the errors.

4.2.2.3 Results

The AR, CDR values for them measurements were calculated and tabulated (Table 4.3). Average performance values for the measurers were also calculated for cross comparison of measurement performances between measurers (Table 4.4). In the table, the values in a row are calculated by averaging the column values for all measurements. Table 4.4 also shows the average deviation between the actual and reported numeric sizes. The deviation gives an idea on the results when the results were not passed through verification and where each reported DM would be counted as one cosmic function points (cfp) whether correct or not.

In measurement results, the component detection rates ranged between 52-100 for data movements, and 60-100 for functional processes. The AR, CDR rates were higher for FP when compared to DM components in the cases except for cases SN and TM. After a mapping of measurement results to the measures, the measurement performances were compared at several levels. The first level comparison among the measurers was based on best and worst AR, CDR rates for DMs and FPs. At the second level, the same comparison was repeated but this time, they were based on average AR, CDR rates for measurements that a single measurer performed. Finally, at the third-level, the comparison was based on average rates

Table 4.3: Results of the Analysis

Measurer	Product Name	Reported /Actual Size (%)	Component Type	Correct	Missing	Incorrect	Actual # of Components	AR (%)	CDR (%)
M1	AN	121	FP	33	1	3	34	89	97
			DM	215	9	57	224	77	96
	BN	109	FP	32	0	2	32	94	100
			DM	159	1	16	160	90	99
M2	MN	84	FP	45	8	0	53	85	85
			DM	211	40	0	251	84	84
	KA	83	FP	45	9	0	54	83	83
			DM	217	45	0	262	83	83
	GH	85	FP	30	6	0	36	83	83
			DM	154	28	0	182	85	85
M3	CN	105	FP	18	1	0	19	95	95
			DM	94	16	21	110	72	85
	SN	127	FP	10	1	0	11	91	91
			DM	60	2	19	62	74	97
M4	TN	126	FP	27	0	0	27	100	100
			DM	128	0	33	128	80	100
	TM	104	FP	3	0	2	3	60	100
			DM	99	3	7	102	91	97
	HB	100	FP	10	0	0	10	100	100
			DM	206	1	0	207	100	100
M5	TCL	164	FP	10	0	2	10	83	100
			DM	72	8	59	80	52	90
	DVTCL	171	FP	3	0	0	3	100	100
			DM	28	0	20	28	58	100

for functional components (without distinguishing DMs and FPs) and on the average values each measurer performed for all products. From this multi-level analysis we could not come up with any significant finding regarding the association between AR and CDR rates and the measurers. In general, the best and worst values were observed to be scattered among the measurements by different measurers and when two measurers' rates are compared, one that has a better CDR rate may have lower AR and vice versa at various levels.

Table 4.4: Measurers Performance Summary

Measurer	DM		FP		DM and FP		Average (Reported / Actual Size) (%)
	AR (%)	CDR(%)	AR(%)	CDR(%)	AR(%)	CDR(%)	
M1	83	98	92	99	88	98	15
M2	84	84	84	84	84	84	16
M3	73	91	93	93	83	92	16
M4	90	99	87	100	88	99	10
M5	55	90	92	100	73	95	68

COSMIC method assigns one cfp for each data movement and when the reported numbers are used without verification, this can easily hide the erroneous concept identifications. This can be illustrated by a comparison of cases CN and TM. In both cases, the reported sizes were within ±5% of the actual size which seems to be quite acceptable. However, case CN has an AR value of 72% where TM has 91% which indicates that CN measurement results include more errors and are less reliable then TM results. This situation is explained by the size contribution of hidden errors (incorrect DMs) in each measurement. On the average, the hidden errors constituted 16% (max.45%) of the reported sizes.

The findings obtained from an analysis of how and why the errors occurred in the measurements were grouped and given from two aspects: functional process based errors and data movement based errors.

Functional Process (FP) Based Errors. After the analysis of the errors, the error patterns in Table 4.5 were found to represent most of the errors made in the measurements. Although requirements allowed in almost half of the cases, we did not observe any occurrence of the other potential error patterns in the checklist.

Table 4.5: Error Patterns Observed in Measurements

#	Error Patterns	The measurements in which the error pattern was observed				
		M1	M2	M3	M4	M5
1	List Before Update/Delete is ignored	-	-	N/A	-	N/A
2	Retrieve Before Update/Delete is ignored	-	-	N/A	CN,SN	N/A
4	Retrieve is defined as a part of another FP	AN	-	-	CN,SN	N/A
5	Transient data group concept is ignored	AN, BN	MN,GH,KA	N/A	SN	-
6	Measurement of other types of operations as a separate FP such as Close, Save.	AN, BN	-	N/A	-	-
7	Multi Pages are considered as separate FPs	-	-	TM	N/A	TCL
9	Parameter tables are considered as OOIs	AN,BN	-	-	CN,SN,TN	-
10	Different Error Messages are considered as seperate Exits	-	-	-	-	TCL,DVTCL
11	Attributes are considered as Data Groups	-	-	-	-	TCL,DVTCL

As per the definitions of COSMIC FSM, a FP is comprised of unique, cohesive and independently executable set of data movements. It is initiated by a triggering event and completed when it has executed all in response to the triggering event. Typically, software functional requirements specify the data needs and relevant software interactions of the software user and they include statements regarding the creation, retrieval, listing and updating the user data. These statements can neither be identified as FPs outside their context nor does their existence imply that they should be valid FPs. However, in the cases where they are valid FPs, they were observed to be open to typical measurement errors. Error patterns 1, 2 and 4 address the failure of detecting the valid ”list” and ”retrieve” FPs that were described in a functional requirement typically to be executed before ”delete” and ”update” functional processes. In patterns 1 and 2 they are ignored and in 4 they are considered as a sub-set of data movements in the functional process typically executed after them. In the cases that included similar requirements, the measurement results did not contain patterns 1, 2 and 4 with the exception of AN, CN, SN cases with small impact on the overall accuracy and error rate. A functional process can be executed following a set of cohesive user actions. A typical design choice is implementing actions in several multiple graphical user interfaces and to be used step-wise due to constraints such as too many input fields to fit in a screen or for step-wise validation of entered data. The functional size and valid functional components should not change with respect to different design choices. In measurements results of AN and BN, measurer 1 identified actions in a multipage form as FPs where they are not functional process themselves; hence does not fulfill a certain response as required by the triggering event. Pattern 6 did not exist in the checklist and was observed in cases AN and BN. It refers to incorrect identification of save, close, cancel actions as FPs. All actions that are made available to the user via controls and interfaces do not necessarily trigger a functional process and when AN, BN results are analyzed we observed that measurer 1 correctly evaluated actions that has no relation with data requirements such as screen navigation actions. However, when the actions were described in requirement statements that emphasize data requirements and along with a valid functional process then they were qualified as functional process mistakenly.

Object of Interest (OOI) or Data Movement (DM) Based Errors. Transient datagroups do not survive beyond the execution of a functional process and they can be derived from persistent data. A valid data group moved in a DM can be transient with respect to COSMIC software model. COSMIC rules imply that they should be identified as a part of a separate object of interest although they may include shared attributes with other persistent object of interests. Pattern 5 refers to errors made in identification of transient data groups. The errors that fit this pattern occurred in ten results measured by four measurers and they were not observed in two measurements only performed by measurer 5 who was aware of the ”transient data group” concept. In DMs where the transient data groups were ignored by the measurers, usually there was a corresponding data group that is associated to another persistent OOI that contains all or a subset of the data attributes of the transient data group, hence the reported size of the software did not change because of the contribution of one data movement; however, the COSMIC rule was violated. Pattern 9 we investigated in the case studies refers to the problem of interpreting ”system parameter” data as data groups and including related data movements in functional processes. COSMIC manual and measurement guides explain how to handle these types of data and when the error is repeated breaking the measurement principles, it systematically leads to overestimation of functional size. The pattern was observed in five of the cases and it was introduced as expected where sets of system parameter attributes were considered as valid OOIs and associated data groups were moved in FPs. Most of the incorrect data movements in cases AN and TN were associated to this pattern and the errors' impact on the results type was relatively high. In cases TCL, DVTCL measurements error messages that are generated as a result of different conditions in the execution of a FP were evaluated as separate data groups and redundant exit data movements were reported. An execution of a functional process may follow different processing paths hence different error or warning messages may need to be delivered to the functional user. As explained in COSMIC manual and guides, the confirmation and error messages should be considered as a single data group and subsequently in all error and confirmation data movements. In addition to obvious errors of this type, we came across with a more confusing situation in which the error messages were not emphasized in the software requirements document. In most of the cases, measures had the tendency to measure the error messages even if they were not specified. On the other hand, few of the participants strongly followed the requirements document and did not measure additional error/confirmation messages where they would be appropriate. We considered such extra data movements as a result of subjective evaluation of requirements and did not mark them as errors in the analysis. According to COSMIC definitions, a data group consists of one or more data attributes. A data movement may move any attributes of a data group. In cases TCL, DVTCL we detected results that contradicts this definition where each attribute of the same data group was recorded as a separate data movement in a functional process. This definition is one of the elementary definitions of the method and was considered one of the simplest rules to understand and apply. However, in the cases the data group includes multi-valued attributes and the measurer evaluated them as separate data groups kept in different database tables. COSMIC meta-model is defined at the logical level and not at the physical level. We associated the introduction of this type of error to the software development background of the measurer.

4.2.3 Case 2: Multiple-case study on Movie Manager Application Requirements

The study was designed to observe measurement performances and errors in a defined context. We defined a set of detailed requirements to isolate inconsistencies in measurement results that associate to high-level, ambiguous or incomplete requirements. The measures had received the same training on the method and had the same level of measurement experience.Another concern in the case design was exposing the measurers to a requirement set that would allow and stimulate the occurrences of the errors and the error patterns under investigation. All of the previously identified functionality properties which were considered as potential sources of measurement errors and challenged the measurers in measurements were artificially seeded into the requirements so that the results would let us discuss their validity through case comparisons. Furthermore, we would be able to observe and analyze the causal dependencies between functionality types and inconsistencies. Accordingly, the requirements specification included cases that involve subject domain properties which include

- complex objects
- inheritance relationships,
- non-persistent (transient) objects,
- constant objects whose state remains unchanged in the entire software life-cycle

and that involve a software behavior described in terms of

- Multiple-step scenarios and actions
- Form/Navigational Elements
- Conditional behavior where software action changes depending on the details of the information input (e.g., different error/confirmation messages for described cases, different business rules for different properties of information entities, etc.)

4.2.3.1 Conduct of the Case Study

We defined a requirement set that is written in natural language and supplemented with an Entity Relationship diagram (Appendix C).It describes the functions of a Movie Manager (MM) database application that covers maintenance of movie related data. Fifteen participants were selected from Software Project Management course students of the Software Management graduate programme. They attended COSMIC v3.0 method training similar to the first case study and had the same level of measurement experience. Different from the first case study, the selection criterion of the participants was a minimum of 80% course attendance. The measurements were conducted at the same place and time. Participants were first given a briefing about the case study and the analysis procedure to be applied. A short question/answer session on the requirements was performed in order to decrease the subjective interpretations and ambiguities in the requirements. Then, the students measured Movie Manager application in two-hour time and recorded the measurement results on a template that includes the same details with the first case study (Appendix A). Two of the participants measurements were not included into the analysis. After measurements were completed, a discussion session was performed in order to get feedback from the participants on the requirement types, which were hard to measure and to see if there were any errors that can be explained as subjective interpretation of the requirements. The case session was recorded on audiovisual media for further analysis and to prevent loss of information.

4.2.3.2 Results

After running the analysis procedure, the errors were distributed and measurement performances were calculated (Table 4.6) (Appendix D).

Table 4.6: Measurement Results and Distributions

Participant	Correct		Incorrect		Missing		AR (%)		CDR (%)		AVG (%)		Reported Size
	FP	DM	FP	DM	FP	DM	FP	DM	FP	DM	AR	CDR	total cfp
1	5	35	0	14	7	47	42	27	42	43	34	42	60
2	11	56	1	26	1	26	85	52	92	68	68	80	100
3	11	66	2	12	1	16	79	67	92	80	73	86	95
4	7	42	0	10	5	40	58	34	58	51	46	55	63
5	7	46	0	6	5	36	58	39	58	56	49	57	63
6	10	57	0	4	2	25	83	53	83	70	68	76	74
7	12	48	1	5	0	34	92	41	100	59	67	79	65
8	10	47	1	2	2	35	77	40	83	57	59	70	60
9	8	37	0	9	4	45	67	29	67	45	48	56	56
10	10	44	3	23	2	38	67	37	83	54	52	68	82
11	10	52	2	31	2	30	71	46	83	63	59	73	101
12	12	57	1	18	0	25	92	53	100	70	73	85	91
13	12	65	3	28	0	17	80	66	100	79	73	90	113
Descriptive Statistics													
Min	5	35	0	2	0	16	42	27	42	43	0.37	0.43	56
Max	12	66	3	31	7	47	92	67	100	80	0.71	0.82	113
Avg	9.62	50.15	1.08	14.46	2.38	31.85	73	45	80	61	0.55	0.64	79
Std.Dev.	2.22	9.75	1.12	9.82	2.22	9.75	0.15	0.13	0.18	0.12	0.1	0.12	0.19
Rel.Std.Dev	0.23	0.19	1.04	0.68	0.93	0.31	0.2	0.28	0.23	0.19	0.18	0.19	0.25

The AR ranged between 42-92% and 27-67% and the CDR between 42-100% and 43-80% in identification of functional processes and data movements, respectively. The average AR and CDR were 73% and 80 % for FP. The average AR for DM was 45% which mean that measurement results included approximately only the half of the data movements that should be identified in the measurement. Average CDR of DM explains that only 61 % of the data movements available in the functional requirements were detected. As the values indicate, the overall performance in identification of FPs was higher than DMs. If the results were to be used without verification, the number of DMs would be reported as the size value and 79% accuracy would be reported on the average. However, when it is compared to the average DM CDR of 61%, the impact of hidden errors become visible. On the average, the hidden errors (incorrect DMs) constituted 21% (max.37%) of the reported sizes. The patterns and their impact in terms of data movement units are given in Table 4.7. The counts in the table are considered to represent the magnitude of the effect of each pattern to the inconsistencies in results. In the table, other incorrect and missing data movements indicate the inconsistencies that could not be explained.

Table 4.7: Error Patterns and Distributions

		Distribution by Participant No												
#	Pattern Description	1	2	3	4	5	6	7	8	9	10	11	12	13
1	No ”exit” for query results	5		2										
2	Exits for populating dropdown form boxes are ignored	2	2	2	2	2	2	?		2		2		
3	Missing Triggering Entry	2				1								
4	List and Retrieve Combined	8	3						6	6				
5	Retrieve and Update Combined				3	2								
6	Query and Detail Listing Combined				3	4	5			2	6	6		
7	Used only parent types	3	4			6		8	12	14	8		13	3
8	Cascading Delete is ignored				9			12		14		?		
9	”Read”s for cascading deletes are missing	7	8	8	4	8	8		8		8	4	7	7
10	Missing Exits for Error/Confirmation	2	3	3	5	4	4	4	3	3	3	4	3	4
11	Other Missing DMs	18	6	1	14	9	6	10	6	4	13	14	2	3
12	Redundant FPs/DMs for conditional cases	1			1					2			2	
13	Assumed a retrieve FP before delete/update FP			3						1	9	25		3
14	Measurement of other types of operations such as Close, Save.													3
15	Used only Sub-Types	4	8			1	1		2	2	3	15	11	16
16	Parameter tables are considered as OOIs	2		4	2	2	2	2	4		2	2		2
17	Different Error Messages are considered as separate Exits	2												
18	Redundant Exits for Error/Confirmation	2		1	2	1		2			1	1	1	2
19	Assumed read before write		?		?									
20	Transient Datagroups are ignored	?	?	?	?	?	?	?	?	?	?	?	?	?
21	Other incorrect DMs	3	18	4	5	2	1	1		4	8	2	4	2

? : The count of error occurrences could not be determined since their occurrences were observed to be coupled with other error patterns in the list.

Functional Process (FP) Based Errors. The errors (patterns 4, 5, 6, 13) in the identification of functional process from statements explaining the creation, retrieval, listing and updating the user data were observed in most of the results and they were associated to both incorrect and missing data movements. For example, 70% of the measurers unified ”query” and ”detail listing” operations as 1 FP although the requirements address seperate FPs. It could be easily concluded that measurers tend to unify FPs that were depicted as consecutive operations in the requirement text. The assumptions on the reuse of an entire FP or a sequence of DMs in the FP were also associated to these errors where examples were observed in the combination of query and list FPs. The patterns 12 and 17 were observed due to the incorrect break down of FPs that involve conditional flows where conditions lead different processing paths on different executions of the functional process, e.g, the definition of the ”Delete Person” operation dictated different operations in case the person to be deleted is a ”director”. Participants mistakenly added a FP as ”Delete Director” or ”Check if Director”. The triggering events for FPs were identified differently by almost every measurer. Since triggering events were represented by a single Entry, they did not contribute to the differences between the numerical results of the measurements. However they were coupled with FP identification error patterns.

Object of Interest (OOI) or Data Movement (DM) Based Errors. The participants identified faulty data movements based on the movement of parameter tables (pattern 16). System parameters were considered as object of interests (OOIs) by 70% of the participants and associated data movements were incorrectly reported.

The requirements associated to showing error and confirmation messages were one of the sources that cause errors and discrepancies in the results. All participants ignored DM at least once that shows the error/confirmation messages (pattern 10). Majority of the participants recorded error message data movements in functional processes although it was not specified in the associated requirements (pattern 18). In the discussion session held with the participants after the case measurements, the participants explained that they assumed that every functional process, by default, should return an error/confirmation message even it is not specified. This type of errors were also caused as a consequence of considering different messages as different data groups and recording redundant exit data movements. Another group of errors (patterns 7, 15) was related to hierarchy of OOIs.According to the guideline for sizing business applications [75],sub-types are the specialized entities that are in the lowest level in the inheritance hierarchy. As it is indicated in the COSMIC guideline, when there is a need to distinguish more than one sub-type in the same functional process, each sub-type is taken as a separate object of interest. The contribution of the OOI on the functional size happens by including additional data movements for each of the special entity (sub-type object of interest) in the functional processes. Movie Manager Application maintains information about directors, scenarist, and cast. Each of these entities has different attributes, thus requiring the identification of sub-entities besides the ”person” general entity. Nearly all of the measurers made mistakes while identifying data movements for sub-types. Measurers used either the parent OOI throughout the whole FPs or its sub-types. Whereas, they are expected to use the parent entity as the OOI when the operation was performed without any reference to sub-type and use sub-entities as OOIs when the sub-types were taken into consideration separately.

Two participants missed exit data movements which return the results of a query to the functional user (pattern 1). Similarly, the exit data movements that populate dynamic dropdown menus with values from valid OOIs were ignored by the majority of the participants (pattern 2). Two participants ignored the triggering entry data movements which are mandatory for a valid functional process (pattern 3). We could not explain why these errors (patterns 1, 2, 3) were made. Another frequent mistake was ignoring the deletion of associated but independent OOIs as a consequence of deletion of an OOI (patterns 8, 9).

One other observation on faulty application of the measurement rules was about software layers which are important in identifying measurement scope and valid data groups and data movements. In functional processes, two participants assumed a Read DM before a Write DM included in an update operation (pattern 19). Based on this observation we deduct that database management operations were taken into account. In other words, they incorrectly counted the operations which are handled by the database management layer, although the layer being measured was the application layer and the data movements in other layers were not to be counted.

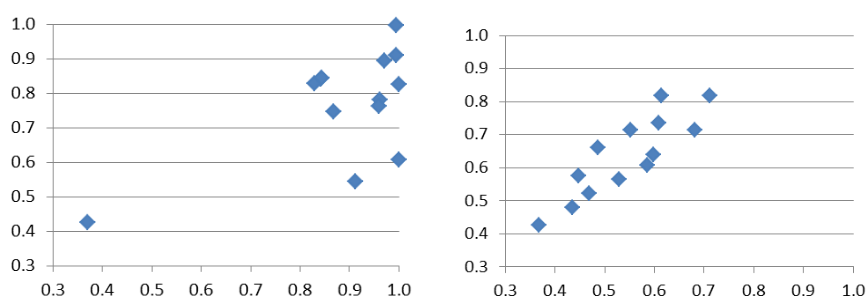
4.2.4 Discussion on the Results of Exploratory Case Studies

The effect of measurement errors on the reliability of the measurement results?

As shown by several measurements observed in both studies, since the errors will be hidden in a single numeric result, it will be misleading to assess the reliability of COSMIC measurements relying on the functional software size values. Therefore, we evaluate the impact with respect to measurement results using the AR, CDR indicators we define.

The weighted averages of CDR and AR performance values represent the performance for a single measurement in terms of functional components (FP and DMs). Although, in the first study the measurements may be considered successful (1 >CDR > .83) in detecting the components (FPs and DMs) in the requirements, the missing or incorrectly identified components overshadowed the success in several of the cases and decreased the performance introducing errors up to as many as the correctly identified components (1>AR> .53).

In the second study, the movie manager application, was measured by different measurers and the weighted measurement performance values were .71>AR>.37 and .82>CDR>.43. Given that the measurement background of the participants are similar and other important factors that may lead to discrepancies in the measurement results are isolated by the case design, it would be meaningful to evaluate the effect of errors relying on the average terms (AR (mean=0.55, stdev=0.10), CDR (mean=0.64, stdev=0.12)). The average AR value indicates that only 55% of the components were reported in the measurements were valid and CDR rate indicates only 64% of the components were detected. The performance measurements indicated that the errors critically affected the reliability of measurement results. Figure 4.2 illustrates AR, CDR plots for the first and second case studies. When they are compared it can be interpreted that the measurement results of the first study are relatively successful in detecting the items from the requirements with high CDR values. The second study results were not as successful and the detection performance was lower. The CDR values were more dispersed in the second study which shows that the measurers were less consistent in detecting the correct base components from the requirements.



(a) Case Study 1 (b) Case Study 2

Figure 4.2: ACR and CDR Rates for Individual Measurements

The accuracy of measurements in first case study measurements were dispersed and were not as successful as in CDR, however they were more accurate when compared to the second study results. The main factor we consider to explain the lower performance of the second study measurements is the measurement challenges. The second study was designed to include diverse type of functional requirements. The abstraction of requirements into functional processes and data movements would require a complete command on COSMIC method principles, rules and examples given in the method manual and guides. Therefore, the measurements were relatively more challenging than first case study products where the challenging conditions and potential error sources were less dense. Another factor could be that the measurers in first case study performed the measurements in several weeks duration and since they performed the measurements as a part of their graduation project they were able to meet specific challenges earlier and were able to get feedback from their advisors on measurement with COSMIC method. On the basis of these characteristics, the performances in measurement results of the second case study can be accepted as a reference lower bound for reliability of measurements using COSMIC method.

The measurement error patterns?

It can be expected that the same measurer makes similar measurement errors or interpretations while constructing models for different sets of functional requirements. For this reason, in the first study, we decided that a pattern should be considered common if it is observed in different measurement results by different measurers. Similarly, the patterns in the second case study were not added to the pattern list if they are not reported in at least two different measurements.

Table 4.5 and 4.7 list repeated errors as patterns across measurements and Table 4.8 gives a summary of error patterns and the associated measurement concepts.

In the first case study, the observed relations between the errors and their identified sources were consistent with the checklist such that we have found evidence that supports the validity of the patterns that were discussed in previous works[27][26] and COSMIC guides[75]. However, the requirement types were sparse in real-life case products and it was not possible to observe occurrence of errors that were in the checklist in all case products. The varying level of detail in the requirements and the different requirement set for each case were the major constraints to develop powerful propositions on the commonality of the errors. The second case design relaxed these constraints by providing a wider range of requirement types and exposing the same and detailed functional requirements set as the measurand. We were able to produce in-depth information about the patterns, cause-effect relations between requirement properties and inconsistencies. Also we were able to associate COSMIC concepts, rules and principles to the errors and requirement properties. The errors of the same type were repro-

Table 4.8: Error Patterns and Associated COSMIC FSM Model Concepts

Pattern	Related Measurement Concept
”Read”s for cascading deletes are missing	OOI
Assumed a retrieve before delete/update	FP
Cascading Deletion is ignored	OOI
Exits for populating dropdown form boxes are ignored	DM
List & Retrieve Combined	FP
Missing Triggering Entry	FP (Triggering Events)
Missing Exits for Error/Confirmation	DM
Multi Pages are considered as separate FPs	FP
No ”exit” for query results	DM
Parameter tables are considered as OOIs	Functional User, OOI
Query & Detail Listing Combined	FP
Redundant Exits for Error/Confirmation	FP,OOI
Redundant FPs for conditional cases	FP
Retrieve and Update Combined	FP
Retrieve is defined as a part of another FP	FP
Transient data group concept is ignored	OOI
Used only parent types or Sub-Types	OOI
Read DM before a Write DM	Layer

duced in different measurement results and were consistent with the checklist and with our previous experience.

4.2.5 Limitations of the Exploratory Case Studies

Construct Validity. Both studies had a multiple-case design hence our findings are based on multiple sources of evidence. The measurement keys were compared to the reported results for detection of the errors in both studies were prepared by COSMIC certified experts and reviewed by other experts and researchers including a member from COSMIC Advisory Board. Hence, the discrepancies were identified in terms of errors from the viewpoint of method experts. In both studies, we applied the same procedure for recording measurement data including details at the same level of granularity using a standard template. Similarly, the same indexes of AR and CDR were used for evaluation of measurement performances and the impact of errors on the reliability in both case studies.

Threats to External Validity The small functional sizes of the industry cases (between 28 cfp and 262 cfp) and the movie manager application (82 cfp) limits the generalization of the results to large size projects. However, the industrial and the Movie Manager software requirements included a variety of functional types and complexity in the requirements. The case study products were business applications and the findings and the error patterns were from the same application domain. COSMIC FSM can be applied to applications in real-time domain and generalizations of our findings to real-time domain require future work. Given the fact that the measurers in both studies did not have a long measurement experience, the generalizability of the findings may be considered to be constrained to COSMIC results by inexperienced measurers. However, all but one of the second case study participants entered and passed COSMIC Entry Level Certification exam that was administered one week after the case had been conducted.

4.3 Application of the 4FSM model

Design in software engineering methods is an iterative and incremental endeavor and the method as a design artifact is effective when it satisfies the requirements in the domain bounded by constraints. Therefore, the design artifact must be rigorously presented using well-designed evaluation methods and the evaluations should give feedback, which is essential in improving the design [139]. Having designed the software functional model for FSM, in this section, we present two empirical studies for the evaluation of 4FSM model. In the first case we perform a retrospective analysis on the measurement results obtained in exploratory case study 2 and evaluate if results would be improved if 4FSM had been used as the meta-model. The second validation study is a multiple-case study where we embed 4FSM in a COSMIC measurement process and analyze the measurement results obtained by its application on a set of requirement specification. We compare the results of this study to the case Study 2 which has a similar setting but has COSMIC measurement results.

4.3.1 Case 3: A Retrospective Analysis of Movie Manager Application Measurement Results

In case study 1 and 2 we explored measurement problems and particularly analyzed the relation between meta-model constructs and inconsistencies in measurement results. The analysis of the measurement results revealed that the majority of the inconsistencies were introduced in COSMIC model construction phase of the measurement process (step-2 in Section 1.1.2).

It was also observed that the significant portion of the inconsistencies could be grouped as patterns that associate to a set of challenging properties of functional requirement specification. In this validation study, we perform an analysis of these challenging properties of specifications and the identified patterns from the perspective of 4FSM model context, constructs and associated rules.

4.3.1.1 Case Design

The aim of this study is to analytically evaluate the ability of 4FSM model in the elimination of some of the previously observed inconsistencies and their sources. We have the following research question:
RQ: How the measurement results would improve if 4FSM was used in case study 2 specifications?

In the literature such a validation design is called as retrospective validation .In contrast to the prospective studies where the validation is performed against the problem, in retrospective design the validation is against the expert evaluation. Retrospective validation is particularly used in software engineering to establish evidence that a design artifact such as a model, method or system does what it intends to do based on a review of historic information. Retrospective validation helps to show how application of a model could have led to different results or conclusions, thus, they can be conducted to answer ”what-if” questions. [140].

For more precise analysis results and to establish a reliable baseline for comparisons we take the Movie Manager case used in case study 2 and explain how challenging patterns would potentially be improved with 4FSM. Other reasons for selection of case study 2 measurements are (1) the availability of detailed information on the measured requirement specification (Movie Manager) and the detailed measurement results, (2) the definition of the context in the case such that several other factors that can potentially cause discrepancies in the results were isolated and (3) Availability of a COSMIC-4FSM Mapping (Table 3.1), thus, having a common unit for comparisons and impact analysis. Accordingly, 4FSM functional message capability (FMC) and stimulus-response pair concepts are conceptually equivalent to COSMIC data movement(DM) and functional process(FP) concepts, respectively.

4.3.1.2 Conduct of the Case Study

Case Study 2 results include a list of error patterns that are expressed in relation to properties of the specifications of the case product MM. Following this information, errors in Case Study 2 were grouped with respect to the several properties of Movie Manager specifications and put in a compact form (Table 4.9). Then for each property a what-if scenario was run and potentials for elimination of inconsistencies were identified and justified. The 4FSM model constructed for MM specifications were used as the reference 4FSM model (Appendix E).

Table 4.9: Properties of Movie Manager Specification and Associated Error Patterns

Specification Property	Associated Error Patterns (from Table 4.7)
Inheritance Relations & Transient,Complex Objects	7,8,9,15
Constant Objects	16
Multiple-step Descriptions	4,5,6,13
Conditional Behavior	12,17,18

4.3.1.3 Analysis Results

MultiStep Descriptions. The error patterns in this group relate to the MM requirement specifications given in terms of scenarios or user operations (e.g., use cases) that are completed in multiple-steps. 85% of the measures could neither consistently decide nor verify whether each individual step should be identified as a separate functional process or they should be combined such that the combined form becomes consistent with the COSMIC FP definition: ”It is complete when it has executed all that is required to be done in response to the triggering event.” For example, a group of measurers interpreted the initial user decision ”update movie information” as the triggering event and identified the scenario List Movies→ Retrieve Movie → Update Movie as a single ”Update Movie” FP. Another group considered List as a separate FP that is ”re-used” in other functional processes and identified Retrieve Movie → Update Movie as another FP. Others have considered each step as a seperate FP. COSMIC FP definition does not reify the concept of FP but it includes the properties that a functional process should have. However COSMIC FSM, attempts to fill this gap by analyzing examples and giving additional rules for FP identification in guidelines [75, p.30].

According to 4FSM, software behavior such as List ,Retrieve and Update would naturally map to stimulus-response pairs in the MM case where the specification explicitly refers to stimuli and the desired responses. In the MM case, each s-r pair is identified as a separate chunk of behavior and thus each of List, Retrieve, Update, Query and Detail Listing are mapped to a separate FP in a COSMIC measurement with 4FSM. 4FSM stimulus and response definitions are generic and do not require any further reference to examples and rules for completeness.

Furthermore, 4FSM is based on the idea that there may be more than one software behavior that satisfies the same of set of requirements and an accurate measurement necessitates a valid design of external software behavior. Thus, the measurement process with 4FSM should not include the process of inventing and specifying a software behavior. Therefore specifications already establishes the relation between observable software behavior with the events in the environment. An external event can be responded by a single s-r pair or a chain of s-r pairs where the ordering of steps can vary with respect to the situation when the event happens and each s-r in the chain may be regarded as an intermediary step of a scenario. This view is also consistent with the definitions and rules given in COSMIC guidelines[75, p.30].

Inheritance Relations & Transient, Complex Objects.MM subject domain data has complex properties. As per the specification of MM, A Movie is a complex object that has simple attributes of title, year of production, production company, genre and set-valued complex attributes of director, producer, writer and cast. During the measurements this complexity challenged almost all of the measurers in several ways. While entering,writing,reading and displaying movie related information, some measurers assumed a single data group that comprises all data attributes, others identified a separate data group for each relation. 4FSM model groups data according to the relations. It further distinguishes internal and external presentation of data. Each FMC should be expressed in terms of flat relations and set valued attributes should be identified as seperate data groups. In Figure 4.3 the internal presentation of all Movie related information is organized into four 3NF relations of the Surrogate Data Model (SDM) component of the model. MM specifications include the details for identification of a transient Movie relation with respect to external presentation.

All the functional messages(FM) and associated Update and Capture FMCs are identified accordingly. For instance whenever a state change is required as a part of the response on the ”Director” of a Movie only the Director state is changed and a single FMC is identified or when the transient Movie nested relation is displayed then five Exit FMC are identified each

Movie

Title(PK)	Year of Production	Company	Genre
Psycho	1967	WB	Horror
ET	1986	20thCF	SciFic

Cast

PersonTitle(PK)	Movie (PK)	Actor_Actress (PK)	Character
A.Hitchcock	Psycho	Anthony Perkins	Norman Bates
A.Hitchcock	Psycho	Vera Miles	Lila Crane
A.Hitchcock	Psycho	John Gavin	Sam Loomis

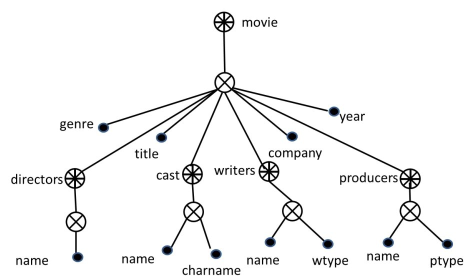
Producer

Producer Name(PK)	Movie(PK)	Type
A.Hitchcock	Psycho	Exective
T.Sheedan	Psycho	Co-Exective

Person

PersonTitle (PK)	Name	Surname
A.Hitchcock	Alfred	Hitchcock
T.Sheedan	Tenesse	Sheedan

(a) 3rd NF Relations



(b) Complex Object Representation

Figure 4.3: Movie Information Internal and External Presentations

enabling passage of a single data group identified from set valued attributes. Similar resolution also applies to the ”person details” complex object available in the MM specifications. The internal presentation of related information is as follows:

- name, date of birth & place of birth,
- movies directed (Title and production year of the movie),
- movies produced (Title and production year of the movie, with co-/executive/producer indicated),
- movies written (Title and production year of the movie, with story/screenplay/story & screenplay indicated),
- movies acted (Title and production year of the movie , with character name indicated)

According to 4FSM external presentation, the ”person details” is identified a nested relation that has set valued complex attributes which are relations themselves. When a response outputs the above data five data groups and associated Exit FMC are consistently identified by detecting four nested flat relations and a top-level relation.

There were also examples for inconsistencies where ”person” entity was interpreted as a generic entity for producer, writer and cast entities (sub-types). For example, only a person data group and associated write data movement was identified while deleting all person details. In SDM Person (Name, PlaceofBirth, DateofBirth) is a separate relation and MM specification imply that a deletion of a tuple in Person relation should be subsequently followed by deletion of tuples in all associated relations. In measurements with 4FSM, the relational thinking instead of Object/Entity thinking resolves such ambiguous cases and the undecidable situations reported in [27] by promoting the selection of FMCs based on relations of set of unique attributes, thus, whenever the attributes that pertain only to a subtype are exchanged, a seperate relation is identified.

A transient datagroup refers to an element of the subject domain for which the software does not keep any state and does not persist any information about. The attributes of a transient data group are calculated by software or derived from persistent information. ”Person Details” and ”Movie Information” are examples from MM domain that are presented to the user as outputs as a result of List Person Details and List Movie Details requests. A common challenge in MM measurements was in the decomposition of transient datagroups into the set of data attributes that the software outputs in a row. In the results several ways of interpretation were observed:

- a single data group was recorded for the whole set referring to all attributes (a.k.a Person Details)
- all or some of the persistent data groups which are required to derive all output attributes were identified as individual data groups

The notion of 4FSM external presentation is to overcome such challenges. Although the relation set of the external presentation may include relations of the SDM, it allows new relations that are transient in nature however are individually recognized and distinguished from SDM relations. ”Person Details” and ”Movie Information” are transient and are represented as complex objects. Their relational analysis results homogeneous sets of data attributes where attributes can be simple or complex. In ”Person Details” case, the identification of movies directed, movies produced, movies written, movies acted as separate relations constitutes a good example for the consistency introduced by external representation notion. Each relation is derived from relational joins of relation ”movie” with director, producer, writer and cast relations. However the output relations are entirely new relations that organizes the transient data. Note that each movies directed, movies produced, movies written, movies acted is a distinct complex attribute and if the requirement specification is altered such that only title and production year of the movie is listed for each complex attribute, the same results will be obtained.

Constant Objects. Constant Objects' state remains unchanged in the entire software life-cycle as per the specifications. In the development jargon they usually correspond to parameter tables, which are the relations that comprise one or more data attributes. COSMIC FSM suggest different interpretations for identification of data groups that represent constant objects. In COSMIC business application guidelines candidate data groups are validated or invalidated from possible views of users (e.g.,”regular users”,”system administrators”) and for a list of scenarios. That is, for the same application the same data group can be a valid and invalid data group depending on the context [75]. For example data groups from parameter tables that are displayed in list-boxes are excluded from measurement since their use is only attribute value selection. Furthermore they are distinguished as second-class data groups which are in support of the ”primary” or ”genuine” data groups. However, other COSMIC resources include examples where the attributes of parameter tables that remains unchanged are considered as valid data groups [141][142].

In the MM case, their identification was observed as a source of inconsistency and ”genre type” of the movies listed in a list-box was considered to be persisted in a parameter table and also identified as valid data group by 70% of the case participants. 4FSM include constant relations in SDM and advocates their identification as valid data groups since they enable proper response calculations and display of information. Since any software behavior involved in validation of the entered data or preventing the entry of invalid data is consistent with the software notion of preventing events. From the external presentation point of view, any relation at the interface can include attributes of the constant relations and they are allowed to be identified as valid functional messages in Enter, Exit, Capture and Update FMCs. However, cases should be distinguished where an identified relation is a constant relation or a relation that include an attribute that has a value-domain predefined by a constant relation.

In the MM case, genre-type is a functional message that populates a list-box and a valid Exit FMC would be identified that moves a functional message ”genre type”. However, while entering the movie information it is used as a domain-value set for the ”genre” attribute of the movie relation and there does not exist another relation ”genre type” that is included in Enter FMC in response to request Add movie.

Conditional Cases. A source of discrepancy in the measurement results was the conditional case specifications which lead to different execution paths, state changes and outputs. Software behavior consequent to an event (or triggering event) that occur in the environment and external to software is determined by the state of the environment and the software. For example in MM, if the person to requested to be deleted has ever directed a movie, the deletion of the person is not allowed, if not then the person is deleted. Several measurers identified seperate FPs for each case while others identified redundant data movements for each different path of execution. Similarly, success and error messages were identified as seperate datagroups for success and error cases. Among others, one observed source of confusion is the same that relate to multiple-step descriptions. It is the FP property given in COSMIC as ”It is complete when it has executed all that is required to be done in response to the triggering event.”. This definition allows the identification of seperate FPs for each different condition and state of the environment. Although COSMIC explains that there are many-to-many relations between triggering events and functional processes, it was not observed to help much possibly due to the reason that FPs are not reified in the COSMIC model by a definition that candidate FPs can be verified against.

4FSM takes this situation into account and defines the response to a stimuli to be determined by the state of the subject domain (environment)and the state of the software. 4FSM distinguishes the stimuli and external events where a stimuli and its response are specification artifacts that are designed to satisfy requirements with the domain knowledge, hence the stimulus-response behavior does not imply a direct relation between external and interface events at any cardinality. Furthermore, 4FSM distinguishes commands from data groups and defines a command as a functional message that comprises several parameters. Thus, a stimulus embodies a mechanism to associate different behavior for different combinations and values of parameters.In an identified s-r pair all the differences in the data collected from the environment, in the state of SDM are handled by the response via suitable FMCs.

In a COSMIC measurement, the size is determined by summing number of data movement types (DMs) identified in each functional process. The results from Case Study 2 include data on the magnitude of the inconsistencies due to patterns (Table4.7). Based on the Case Study 2 results and establishing the relations between functional properties and error patterns, Table4.10 is obtained. The numbers indicate the count of incorrect or missing DMs that could be potentially avoided for each measurer if 4FSM was used in the measurements. On the average a 62% reduction in errors would be possible with 4FSM.

Table 4.10: Improvement Potentials in Movie Manager Measurements

Measurer	1	2	3	4	5	6	7	9	10	11	12	13	14	Average
Total # of Inconsistencies (without 4FSM)	61	52	28	50	42	29	39	41	54	61	75	43	45	48
Total # of Inconsistencies (with 4FSM)	32	29	12	26	18	13	15	9	13	24	22	9	12	18
Error Reduction (%)	48	44	57	48	57	55	62	78	76	61	71	79	73	62

In this study, we have investigated how results would improve if 4FSM model was used in Movie Manager measurements. First, we showed how measurement errors that are causally dependent to a set of properties of MM specifications could be avoided by 4FSM. Our reasoning is based on the formalized concepts and rules of the 4FSM models. We have calculated a 62% reduction in errors.

4.3.1.4 Limitations

Internal Validity. There is not any proven training and guidance document available on 4FSM model. The analysis is based on the presumption that all measurers of Case Study 2 would had all understood the concepts and the rules of the 4FSM model. In error reduction calculations we have assumed that the errors are solely caused by the properties of the specifications, thus 62% indicates a maximum improvement potential.

External Validity. The comparative analysis performed in this case requires complex requirement specifications, detailed information on the subject and the constructed model and an expensive cause-effect analysis. There are no other published empirical data that includes detailed measurement results and requirement specifications. For this reason, the case design was limited to the Movie Manager Application specification. However, the analysis is based on generic functional properties and their resolution with 4FSM. Many of them were seeded into the MM specification. In Case Study 1 they were also observed to cause discrepancies in the measurement results of real-life projects.

Reliability. One problem relating to retrospective validation is hindsight bias, which refers to the ability of people to reconstruct prior probabilities for an event after it has occurred. The analysis and the results of this case depend on the researcher's judgments and decisions. However, the causal relationships between the specification and measurement results have been established with two other PhD candidate researchers during the movie manager case study. Other studies and researchers also establish and simulate similar relations between functional properties and measurement inconsistencies [26][27]. The analysis performed was also reviewed by another FSM expert who is a PhD candidate doing FSM research.

4.3.2 Case 4: Application of 4FSM to an Online Order Software

The goal of this validation study is to evaluate software functionality model construction performances using 4FSM. The evaluations are based on the the qualitative analysis of eight models constructed by graduate students.

4.3.2.1 Design of the Study

The study was driven by the following research question:
RQ1. Is 4FSM efficacious in constructing consistent software functionality models for FSM?

We have selected a holistic multiple-case design strategy for the study. The unit of analysis were set as a 4FSM model constructed by a graduate student for the same set of requirement specifications. Two sources of information were decided to be used for evaluations. First is the results obtained from the comparison of the consistency observed in the 4FSM models to the consistency observed in COSMIC software models in Case Study 2 and second is the measurers' feedback on the selected characteristics of the 4FSM model.

In order to enable comparisons between the models, the case was conducted in a defined context. The set of requirement specifications were prepared synthetically such that they include similar properties with Movie Manager (MM) software application which were found to cause measurement challenges and lead inconsistencies in constructed software models. The specifications include description of a Online Order Software (WOS) that include following behavioral and data properties:

- complex objects
- inheritance relationships,
- non-persistent (transient) objects and
- constant objects whose state remains unchanged in the entire software life-cycle

and that involve a software behavior described in terms of

- multiple-step scenarios and actions,
- form/navigational elements and
- conditional behavior where software action depends on the information collected from the environment and software state.

The specifications were given in natural language and subject domain was described with an Entity Relationship diagram and its presentation in 3rd NF relations(Appendix F). Five PhD and three MSc students were selected as measurers from graduate students at Middle East Technical University, Informatics Institute. A basic knowledge and experience on COSMIC FSM method and relational data analysis was expected from the measurers so that the measurers could give feedback on the model and to ensure similar training characteristics of the measurers in Case Study 2. A measurement report template was prepared such that the model can be represented by Functional Message Capabilities(FMC) utilized by each s-r pair and data groups with respect to internal and external presentations.

In order to get post-task feedback from the measurers, we have defined the high-level characteristics for 4FSM and then prepared a questionnaire that includes 11 questions. Each question associates to a low-level characteristics defined for a high-level characteristic (Table 4.12). Three questions were added for a comparison of the the measurand's (WOS specifications) characteristics with respect to the MM application and the measurer's previous measurement experience. The questions were formulated using a 5-point Likert scale, using the opposing statements question format (Table 4.11).

Table 4.11: Excerpt from the Questionnaire

4FSM model data concepts are intuitive						4FSM model data concepts are abstract

We have also added 3 open-ended question to get a feedback of the 4FSM model with comparisons to other FSM method models students have experience with and their suggestions for improvement.The questionnaire form can be found in Appendix F

Table 4.12: High-Level and Low-Level 4FSM Characteristics

High-Level	Low-level
Level of Abstraction (LA)	Behavioral concepts (Q2) Data concepts(Q1)
Ease of Understanding (EU)	Behavioral concepts(Q3) Data concepts (Q4)
OverAll Model Robustness (OMR)	Ease of Construction (Q6) Adequacy of Guidance (Q7) Consistency by Principles (Q14) Possibility for Verification (Q8) Adequacy of Measurement Procedure (Q5)
Learning Requirements	Adequacy of Training and Documentation (LR) (Q9,Q10)

4.3.2.2 Data Analysis Procedure

The same data analysis procedure approach given in Case Study 2 was followed and for each model constructed by the measurer, the data analysis was performed in two steps. In step 1, the reported s-r pairs were compared against the s-r pairs in the key 4FSM model constructed by the researcher. The matching s-r pairs were detected checking (stimulus statement,response statement, stimulus FMC (command)) components of s-r four-tuples and were marked as correct. If a s-r pair in the key was not reported by the measurer it was marked as missing. When the reported s-r was not matching any in the key it was marked as incorrect. In step 2 the FMCs that the responses utilized was checked. For each reported FMC the matching FMC in the key was compared by checking the functional message (data group or command), the recipient (for Exit FMCs) and sender (for Entry FMCs) components of the FMC. For data groups, the matching relations in external and internal presentations and their types (constant,single attribute,flat,nested) were checked. Different namings for same intended relations (e.g, orderdetails, ordersummary, orderwithtotalcost) were considered indifferent whenever they can be resolved. Finally, following the same convention for s-r pairs, FMCs was recorded as correct, missing or incorrect . A final adjustment was made if it was detected that ”Multiple-step scenarios and actions” were combined into single s-r or ”Conditional behavior” was recorded as seperate s-r pairs. In the former case, the s-r was matched to the s-r with the highest number of FMCs and for the former both were merged and combined and matched to the s-r pair in the key and the step 2 was repeated. Whenever a resolution was required , the measurers were asked for clarification. The models constructed by the measurers can be found in Appendix F.

4.3.2.3 Conduct of the Study

A training document on the fundamental concepts and rules of 4FSM model was prepared and the measurers were given a two-hours 4FSM training. In the training, relational model of data was summarized including examples for flat and nested relationships and attribute domain analysis. At the end of the training, instructions for filling out the measurement report templates were given. Then, the case product specifications,a copy of the training material and the measurement templates were distributed to measurers. An overview of the specifications was also given. The measurers were allowed to ask questions about 4FSM and specifications in their model construction process.No time limit was set for performing the measurements. The questionnaire form were filled after the measurement.

After the data analysis the root cause analysis for missing and incorrect s-r pairs, functional messages (data groups and commands) and functional message capabilities was performed. Whenever it was not clear why the components were not identified or missed, additional information from the measurers were taken for confirmation about their root causes.

4.3.2.4 Results

After performing the data analysis procedure and converting the results into COSMIC equivalents according to the COSMIC-4FSM mapping given in Section 3.2 Table 4.13 was obtained.

In order to check whether results have been improved with 4FSM, the percentage distributions of correct, incorrect and missing FPs and DMs were calculated for all MM and WOS measurements (Figure 4.4). We have also calculated (incorrect+missing)/correct DM ratios for

Table 4.13: Measurement Results and Distributions

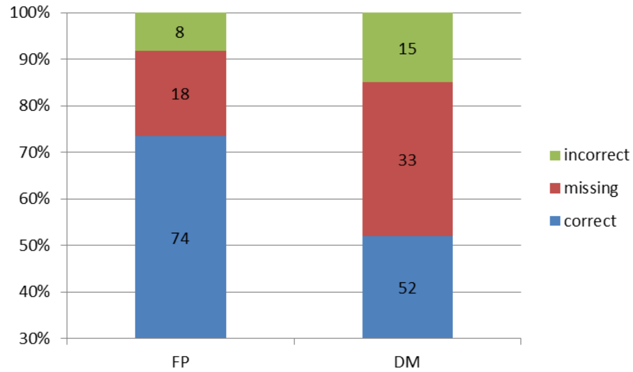
Participant	Correct		Incorrect		Missing	
	FP(S-R)	DM(FMC)	FP(S-R)	DM(FMC)	FP(S-R)	DM(FMC)
1	9	61	0	1	0	3
2	9	61	0	0	0	3
3	9	64	0	1	0	0
4	9	59	2	11	0	5
5	9	59	1	15	0	5
6	9	59	0	0	0	5
7	9	60	0	0	0	4
8	9	60	0	5	0	4
Descriptives						
Min	9	59.00	0.00	0.00	0	0.00
Max	9	64.00	2.00	15.00	0	5.00
Mean	9.00	60.38	0.38	4.13	0.00	3.63
Std.Dev	0	1.69	0.74	5.82	0	1.69

pairwise comparisons of measurement performances in both cases (Table 4.14). Accordingly, the overall performance in the constructed 4FSM models for WOS specifications were observed to be significantly better than the constructed COSMIC models for MM (52% Correct DMs in MM, 89% Correct DMs in WOS). At individual levels, all (incorrect+missing)/correct ratios in WOS measurements were significantly better than every ratio in MM measurements.

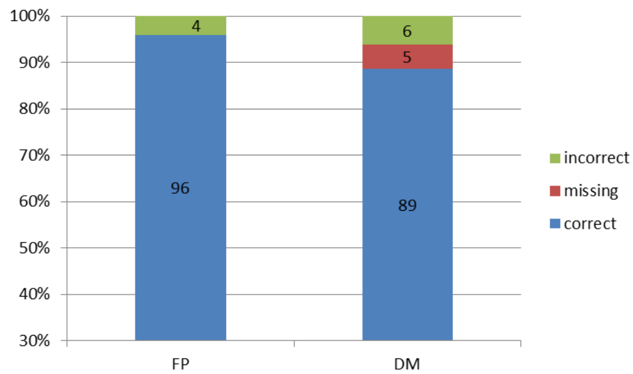
Table 4.14: (Incorrect+Missing)/Correct DM Ratios

	1	2	3	4	5	6	7	8	9	10	11	12	13
MM	1.74	0.93	0.42	1.19	0.91	0.51	0.81	0.79	1.46	1.39	1.17	0.75	0.69
WOS	0.07	0.05	0.02	0.27	0.34	0.08	0.15	0.07					

According to the inconsistency analysis approach we follow, we not only consider the incorrect model components (FPs and DMs) but also the data movements that are expected to be identified (missing). Total number of correct and missing items is a constant that gives the number of components that exist in the reference software model constructed by the researcher and gives the accurate model(Total # of Valid Components=# of Missing Components + # of Incorrect Components). The meta-model of a FSM method should enable the identification of all components while not-permitting the identification of any unintended and invalid components. In this context, the effectiveness of a FSM method meta-model in achieving consistency can be supported by the observation of a negative correlation between the number of correctly identified components and the number of missing and incorrect components. Thus,



(a) Case Study 2

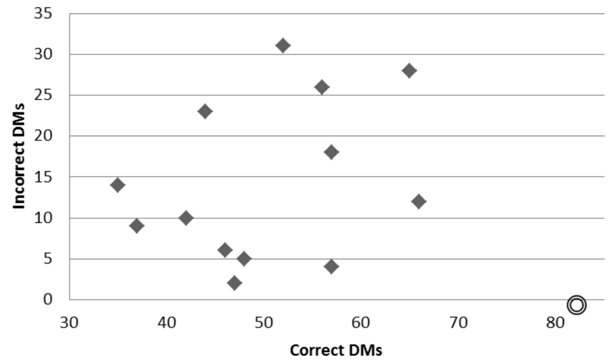


(b) Case Study 4

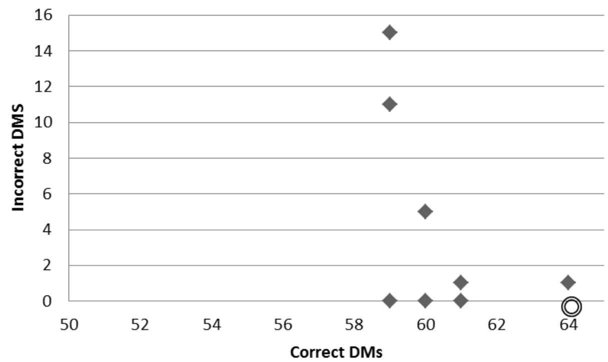
Figure 4.4: Correct, Missing, Incorrect FP-DM Distributions for MM and WOS

the models constructed by the measurers following a FSM meta-model are consistent when a convergence to the correct (intended) model can be observed. Following this, in Figure 4.5 plots for correct and incorrect data movements are given for MM and WOS measurements. The missing data movements in the analysis were not included since it is determined by the number of correct data movements, hence they are not independent variables. FPs were not included in the analysis since the inconsistencies in their identification are also represented in the counts of correct and incorrect data groups.

First, we test whether as the number of incorrect data movements (wdm) identified according to COSMIC meta-model increases as correctly identified components increase (cdm). Both variables were tested for the normal distribution using the Shapiro-Wilk test (Table4.15). The results indicate that both variables can be assumed to have normal distributions ($p > 0.05$). The count measure for wdm and cdm have at least an interval scale and we apply single-tail Pearson's correlation to test our hypothesis (Table4.16). The result shows that there is not



(a) Case Study 2



(b) Case Study 4

Figure 4.5: Correct and Incorrect Data Movements Plots

a significant correlation between wdm and cdm and we can not say that with the COSMIC meta-model the more correct DMs identified the less incorrect DMs observed for the MM measurements ($r = .141$, $p(one - tailed) > 0.05$).

Table 4.15: Tests of Normality for cdm and wdm in MM

	Shapiro Wilk		
	Statistic	df	Sig.
wdm	0.924	13	0.280
cdm	0.961	13	0.775

We could not run the Pearson Coefficient analysis for WOS since we have a very small sample size (N=8). However visual inspection gives us some evidence. First conclusion is that with 4FSM a measurer tend to make less errors as the measurer identify more data movements (or equivalently FMCs) correctly. In the figure, the circles on the plots at coordinates (82,0) and (64,0) indicate the correct number of data movements according to the reference models

Table 4.16: Pearson Correlation Results for Variables cdm and wdm

Correlations			
		cdm	wdm
cdm	Pearson Correlation	1	0.321964
	Sig. (1-tailed)		0.141683
	N	13	13

respectively. The results in the WOS case converge to the point as they move along ”correct DMs” axis.

The WOS specifications include properties that we have found to challenge the measures and lead to inconsistencies in Case Study 1 and 2. 4FSM is expected to control the measurement problems associated to these properties and avoid discrepancies. The same set of properties exists in MM specifications in Case Study 2 and the associated error distributions were given in (Table 4.7). In Table 4.9 Case Study 2 error distributions and properties were re-organized according to the compact form described in Table 4.18. After the an analysis of the missing and incorrect data movements the distribution of errors for WOS measurements are given in Table 4.17 in the compact form.

Table 4.17: WOS Error Distributions in Compact Form

(Type)Property \ Measurer	1	2	3	4	5	6	7	8
(1)Inheritance Relationships& Transient,Complex objects				4	4	2		
(2)Constant Objects								
(3)Multiple-step Descriptions					6			
(4)Conditional Behavior				5	2			
Other Errors								
Input-Output-Relations	3	2	1	3	4		4	9
Attributes of a Relation are identified as Separate Relations	1	1		3		1		
Data Groups and Commands are assumed to utilize same FMC		2	1	2	5			

When the results are compared to MM error costs in compact form, it can be observed that with 4FSM only three out of eight measurers have found inconsistent results for properties of type (1) in WOS where in MM measurements type (1) properties have caused all measurers introduce inconsistencies. Similarly the ratios are (0/8,1/8,2/8) for MM and (10/13, 11/13,10/13) for WOS for types (2),(3) and (4) respectively. Furthermore, the error impact of the properties is significantly less than the impact in MM in consistency with high consistency rates given in Table 4.13 and Figure 4.4. Therefore we find further evidence that in consistency with Case 3 results, 4FSM can be effective in controlling such challenges.

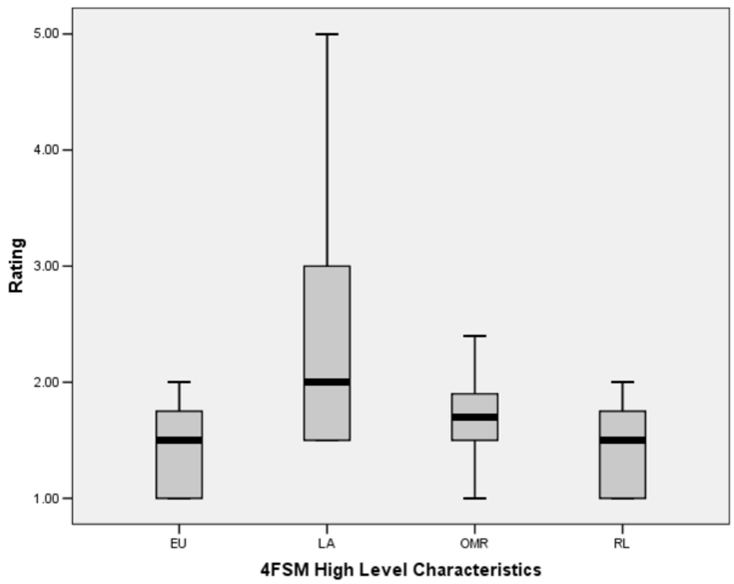
Table 4.18: MM Error Distributions in Compact Form

Measurer	1	2	3	4	5	6	7	8	9	10	11	12	13
(1)Inheritance Relationships& Transient, Complex objects	14	20	8	13	15	9	20	22	30	19	19	31	26
(2)Constant Objects	2	0	4	2	2	2	2	4	0	2	2	0	2
(3)Multiple-step Descriptions	8	3	3	6	6	5	0	6	9	15	31	0	3
(4)Conditional Behavior&	5	0	1	3	1	0	2	0	2	1	1	3	2

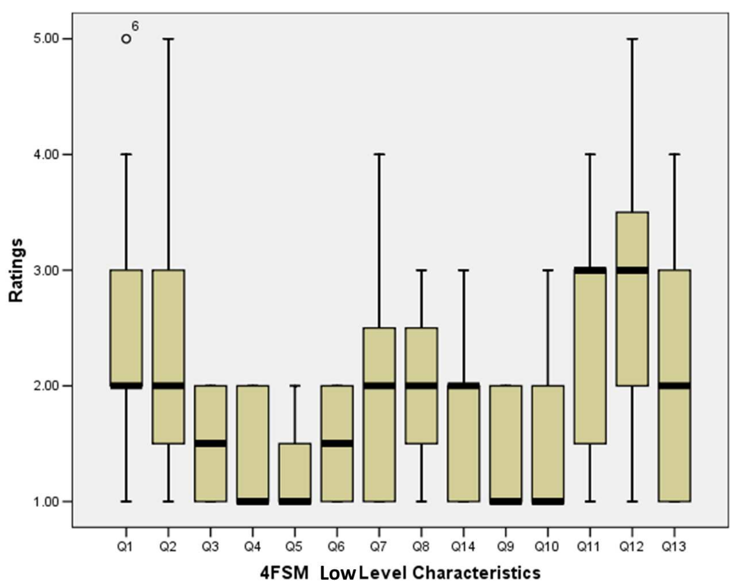
The characteristics of 4FSM as perceived by the users are given in Figure 4.6. The feedback from measurers were analyzed with respect to the high-level (LA, EU, OMR, RL) and low-level characteristics. The high-level characteristics rates for each measures were calculated by averaging the rates given for the associated questions for each measurer. The evaluations were based on comparisons of the results to value 3 which corresponds to the ”neutral” score.

According to the scores, the measurers found 4FSM concepts intuitive without any significant difference between data and behavioral concepts. 4FSM concepts were found easy to understand, however data concepts were found relatively easier to understand. The overall robustness of the model was found good ,however the contribution of easiness of model construction and procedure application to the robustness perception was higher then the guidance and possibility of verification. The training and material given in the conduct was found sufficient for the application of the model. The measurers did not perceive any significant difference in the difficulty between MM and WOS specifications from a measurement point of view (Q15). Besides, they found the difficulty of WOS specifications similar to the specifications they had measured before (Q16). The open-ended questions were intended fro comparison of 4FSM with other 4FSM models the measurers know. Majority of the users found 4FSM to have clear and precise concepts and rules in comparison to other FSM methods. Stimulus-response pairs and associated concepts were found to be the most significant contribution on the basis of challenges met in identification of COSMIC functional processes. The questionnaire results can be found in Appendix F.

The error analysis also provided us with feedback on several shortcomings of 4FSM rules and concepts. The majority of errors in handling properties of type (3),(4) was associated to command concept. Several measurers had difficulty in distinguishing a command and its parameters. For instance in ”query products” behavior specification , 2 of the users considered separate stimulus for the user options of ”query by product property” and ”query by product



(a)



(b)

Figure 4.6: High-Level and Low-Level Characteristics Ratings for 4FSM

title" where the specifications imply that both options are communicated to software via single event at the interface (e.g click query button). The measurers consistently identify FMCs that enable the reception of a stimuli.On the contrary ,the FMCs utilized in the response part of the s-r pairs and that enables passage of commands to external software (e.g. mail server,bank web service) were ignored when they utilize a FMC that passes a datagroup to the same external software. Another frequent interpretation was in failure to distinguish a relation's attribute domain when the domain is shared with attributes of another relation. Four of the measurers did not explicitly identify non-observable s-r pairs and their dependency relation between observable s-r pairs. However they have constructed their models such that the non-observable s-r pairs' response part was merged with those of observable s-r pairs. That is, they have reported the normalized form of their models.

To sum up ,in this validation study, our findings from an analytical evaluation of the models constructed from different sources of information indicate that the application of 4FSM was efficacious in enabling the construction of consistent functionality models.

4.3.2.5 Validity Threats

In this section, we discuss several issues that can affect the validity of the validation study.

Construct Validity. The MM measurement analysis was performed on the basis of error types and their sources in the requirement specifications. 4FSM model design process included their resolution as a design target. In this validation study, evidence on the performance of 4FSM was obtained by assessing the measurements against the same types of errors and their sources in the requirement specifications.

Threats to Internal Validity In this case study, the functionality models were constructed for a software specification (WOS) which is different from case study 2 requirement specifications (MM) . However both specifications has small and similar functional sizes (WOS:64 cfp,MM:82 cfp)and they include the same properties that were found to cause inconsistencies between models constructed by different measurers. The questionnaire results show that the measurers also agree that there was not a significant difference between specifications from a measurement point of view (Q12,Fig. 4.5). Another difference was that the measures of Case Study 2 were different from the measurers of this study. However, the measurers selected in both case studies were graduate students in IS having similar software engineering profiles and had similar levels of FSM knowledge. Besides majority of both group of measurers hold entry-level COSMIC certificate. Another limitation of the study was the small number of sample observations which did not allow statistical analysis, however analytic investigation allowed us to reason how 4FSM could avoid incorrect results from challenging requirements. As a future work the validation results should be repeated with between-subject experimental studies having a larger sample size and using a single specifications in both control and treatment groups.

Threats to External Validity The results we have obtained rely on comparisons between two specifications with small functional sizes. In both cases the specifications were synthetic such that they were rich of properties that challenge measurers to enable observations. However our conclusions rely on the observation that using 4FSM discrepancies due to challenging functional properties are eliminated. Therefore, we can not say whether the performance improvements will be higher or lower for measurements performed on specifications with larger functional sizes as we do not know the frequency of such properties in their specifications. As a future work study should be extended to real-life projects with high functional size values.

Another limitation on generalization of the results is the application domain of the specifications that were measured. They were both Information System specifications. 4FSM is suitable for software whose functionality can be expressed in sequential s-r pairs. However a broad category of real-time systems behavior can be expressed by sequential behavior. As a future work the validation studies should be extended to real-time systems domain.

In the study, we decided to select measurers' with some knowledge on a FSM method so that they can compare the adequacy of 4FSM to the other meta-models in controlling measurement challenges and provide data for evaluation. This constitutes another limitation on the generalization of high 4FSM performances to the measurers having no previous FSM background. Future work should address validation of 4FSM with measurers without any FSM knowledge.

4.3.3 Summary of Validation Studies

In Section 3, we proposed the 4FSM model with formalized concepts,relations and rules in order to facilitate construction of reliable software functionality models for Functional Size Measurement. Our approach to reliability was not limited to the closeness between the numeric values obtained by measurers. We extended the reliability assessment to the models constructed using a FSM method meta-model.

In Case Study 3, we retrospectively evaluated the potential for improvements in software functionality model construction with 4FSM. The analysis units were selected as the COSMIC functionality models constructed by 13 students for the Movie Manager(MM) application specifications. Availability of detailed information on the properties of the application and the constructed models allowed us to evaluate the improvement potentials analytically. In the analysis of properties we have demonstrated how the cases are handled by 4FSM formalized concepts and rules. We have also calculated a 62 % error reduction with 4FSM which indicates a significant improvement in the reliability of the models and functional size values.

In Case Study 4, we trained students in constructing 4FSM models and they constructed 4FSM models for an Online Order System (WOS)that has similar functional properties with MM application. The results of the analysis provided us with evidence that measurers constructed models that are consistent with eachother and that converge to the reference model. Besides they introduced few components that do not exist in the reference model.

When the results are compared to the models of MM constructed using COSMIC meta-model, we found significant improvements in individual and overall results. In the study we have also prepared a questionnaire in order to get students' post-task feedback on 4FSM characteristics. In the questionnaire we have also asked how they compare 4FSM to the other FSM methods they know. The feedback was positive such that they found 4FSM robust, easy to learn and understand and to provide concepts rules for more precise measurements in comparison to FSM methods they have experience with. The results of Case Study 4 also address potential improvement opportunities in 4FSM concepts and rules. The significance of the results of the validation studies mainly relies on an analytic investigation and comparison of the constructed functionality models with COSMIC and 4FSM.

We are aware that the validation studies had several limitations. Further validation with sufficient number of cases to evaluate 4FSM performance with statistical significance is required for direct and powerful validation results. Although they were considered an controlled to an extent, further empirical studies should be designed such that the potential validation bias due to the differences in requirement specifications, measures background, experience should be minimized. As a future work we are planning to design between-subjects experiments with increased number of participants and control groups to address such concerns and perform further validation.

CHAPTER 5

CONCLUSIONS

In this thesis, we have addressed reliability issues in FSM. The nature and the extend of subjectivity in measurements were explored and their impact on the reliability of the measurements were evaluated. The functionality models constructed by measurers were analyzed and sources of discrepancies between models were identified. We have performed a systematic literature survey to understand how formalization in FSM helps FSM practices. Review results revealed that formalization proposals in FSM literature were limited to the definitions of general properties of FSM methods and to the measurement practices for selected semi-formal and formal specification styles. In order to tackle subjectivity in FSM, we have introduced a software functionality meta-model (4FSM) for functional size measurement .

The novel contribution of this thesis study is the 4FSM model that is founded on formalized terms and concepts. It's objective is to enable construction of reliable functionality models.

This chapter summarizes the contributions of this thesis study, addresses its limitations and suggests future research directions based on the findings discovered during the study.

5.1 Contributions

4FSM Model and Validation Results. The major contribution of this thesis study is the 4FSM model for FSM. In the model, the components and their relations are given in formal terms. Besides, norms and rules for ensuring the proper identification of homogeneous components are given. In the design process of 4FSM model, a set of generic properties of software specifications that cause discrepancies were taken into account. We have established associations between 4FSM and COSMIC FSM meta-models and provided a mapping in order to demonstrate how 4FSM can be used in a method and numeric functional size value can be obtained. FSM methods rely on similar concepts [89] and we address establishing associations to widely used IFPUG method as a future direction.

We have performed two cases studies (Case Study 3, Case Study 4) for the validation of the 4FSM. The first validation study have shown that the model effectively facilitates the resolution of the functional properties that challenge measurers in the application of COSMIC FSM method. The study demonstrated that using the the concepts and the rules of the 4FSM meta-model, errors due to functional properties could be avoided and a 62% reduction in errors could be explained. The second validation study demonstrates that 4FSM help measurers in the construction of consistent functionality models and obtaining reliable size measure values. In the study, the comparison of 4FSM results of a specification to the COSMIC results of a similar specification indicates that correct Data Movement percentages increased from 52% to 89%. Furthermore with 4FSM, measures could successfully resolve challenging functional properties, in the specifications. The measures feedback on the model provided us with further evidence that the 4FSM model definitions and rules help them to identify components more precisely than other FSM methods they have experience with. Their ratings of 4FSM characteristics on a five point scale indicated that 4FSM was perceived to be robust(< 2), easy to understand (< 2) and they have found the concepts concrete(< 2). They have also found 4FSM training adequate for proper 4FSM application and also have suggested improvements for the training and clarification for the concepts of the model.

Requirements Engineering View Point. Another significant contribution of 4FSM is the precise characterization of the concept of functionality and associated software artifacts. The software context for 4FSM rely on the foundational concepts of requirements engineering and reactive systems. Such characterization is essential in FSM such that it helps FSM practitioners distinguish the artifacts that pertain to software functionality, determine their measurement scope and evaluate the adequacy of available artifacts for measurement regardless of the FSM method they use. Besides, it provides measurers with a baseline to understand what type of artifacts a selected FSM method quantifies. Moreover, the difference in the notions of requirement specification and functional size measurement (i.e. model construction) has been emphasized and the measurer's role have been refined. This distinction is crucial in that it avoids inconsistency in models preventing measurers assume or devise a software behavior (functionality) solely from requirements or phenomena (events, states) private to the environment software operates in.

Another feedback provided to the FSM community is a collection of generic requirements specification properties that challenges measurers and lead discrepancies in measurement results. Although some guidance for handling several of these properties have been given through examples in method guides[75][143], we suggest that they should be extended. In addition, further reasoning in their resolution should be given in terms of the meta-model concepts and rules.

Formalization Studies in FSM We conducted an extensive survey in FSM formalization literature. We grouped them with respect to their proposes. It revealed that a group of formalisms were introduced for assessing generalized properties of FSM methods and their comparsion. The other group of studies were automation oriented and they explored the automation opportunities for formal or semi-formal specification styles. The survey also addressed how their use can help. One significant contributions of the FSM studies were the evaluation of the adequacy of FSM methods from a formalization point of view.

Other Results of Exploratory Case Studies. A practical and indirect contribution of the exploratory studies (Case 1 and Case 2) conducted at SMRG as part of a research project [138] and explained in this thesis work is the feedback provided to FSM practitioners. In most of the FSM reliability research, results rely on the dispersion characteristics in the numeric results. In the case studies the data analyzed in the reliability analysis was not limited to total functional size values and was extended to data that represent the constructed models. The findings revealed that ”incorrect” components and constructs (such as Functional Processes and Data Movements) can critically contribute to the total size (avg.16%, max.45% in Case Study 1 and avg.21%, max.37% in Case Study 2) with COSMIC FSM. Although the criticality of hidden errors were not explored for other FSM methods and for experienced measurers, we believe this is a vulnerability of FSM practices in the utilization of functional size. We suggest that for a reliable usage of functional size values, constructed models should be checked against the specification and the FSM method meta-models. Besides, other corrective and preventive measures should be taken to control hidden errors and avoid undesired consequences. Similarly, the functional size values in international benchmark and historical data-sets should be used cautiously. A suggested way for verification or prevention is the utilization of FSM experts, however it may be inefficient due to time and budget constraints.

Therefore, we suggest FSM researchers to introduce more formalism into meta-model and model constructions and enable opportunities for measurement automation and verification. We include automated measurement in the directions for future work.

Two M.Sc. thesis studies at Informatics Institute, METU address this issues and the thesis' scopes include models, techniques and tools for detection and verification of COSMIC measurements results on the basis of specification properties and COSMIC method rules[144][145].

5.2 Limitations and Future Work

Specifications Size and Characteristics. The results of Case Study 1 rely on the measurements of 12 different industrial project specifications that have varying but small functional sizes. Case Study 2 project's specification also has a small size. The focus of both case studies was investigation of relations between functional properties and discrepancies and we were able to explain discrepancies in a systematic way. The set of functional properties are generic and can exist in many specification in arbitrary frequencies, thus poses reliability risks in measurement of projects whether small or large. However, we can not say whether the reliability threats will be as critical as for the measurements for projects with larger functional sizes. On one hand small projects are more sensitive to errors; on the other hand large project may include challenging properties in higher frequencies and may include properties we have not met in small projects. Similarly the specifications in Case Study 3 and Case Study 4 had small sizes. As a future work, the robustness of 4FSM model against large size projects with challenging properties may be evaluated performing further empirical validation.

Specification Similarity & Measurer Profiles. The purposes of the validation case studies were the demonstration and assessment of the effectiveness of the 4FSM model. In the second validation study, the performances of 4FSM measurements were compared to Case Study 2 COSMIC measurements. Although the specifications being measured in each case study had similar characteristics and included the same functional properties with similar frequencies, they were different. The measurers profiles between two case studies were very similar in software engineering knowledge and there was not a significant difference with respect to COSMIC expertise criteria [73]. Another limitation was the low number of observations which did not permit an analysis with statistical significance. Further experimental studies with between-subjects design and larger sample sizes can be performed in order to evaluate reliability of the improvements with 4FSM,controlling potential bias from the specifications, measurers' level of expertise in FSM and the environment.

Applicability The applicability of 4FSM relies on the availability of data that is is or can be modeled according to the relational model. 4FSM is well-suited for data-strong information system software and its validation was performed for transaction systems. 4FSM model has been designed primarily for software whose functionality can be expressed in terms of sequential stimulus-response behavior thus it is not applicable to systems where their functionality involves concurrent behavior (e.g multi-player video games). However a broad category of real-time systems behavior can be expressed by sequential behavior. Moreover 4FSM is compatible with COSMIC meta-model and COSMIC is applicable to real-time system software. These provide some evidence that 4FSM can be a promising model for real-time system software and it's applicability should be validated on the basis of measurements on real-time specifications and the feed-back from the practitioners in real-time development environments.

The generic functional properties 4FSM handles are identified from FSM method guides and explorative case studies. They include properties such as complex data structures, transient relations and stimulus-response dependencies and 4FSM's approach is to result in a measurable scheme that has homogeneous components that can be checked against definitions rules and obtained by following the given procedures. However there can be other properties that we have not met. 4FSM provides a formal foundation for modeling functionality and it allows inclusion of further rules and procedures for further normalization.

4FSM Model Concepts &Training Requirements. In the validation studies we have discovered that discrepancies in several measurers' models were caused by different interpretations of the concepts of stimulus,command and command parameters. While other concepts were more similar to COSMIC equivalents, these were relatively new concepts and the training materials and guidance was limited to few examples.The measures have provided feedback that training and documents could be improved. In order to resolve ambiguities the definitions for these concepts need improvement and measurers should be supported with further guidance,examples and enhanced training materials.

Automation Opportunities for 4FSM Model Construction. 4FSM model is constructed from software artifacts that pertain to the externally observable behavior of software and its subject domain. These artifacts are not available only in natural language and are also expressed in selected semi-formal notations or formal languages which are already supported by automated tools. As a future direction, concepts maps between the elements of selected notations and 4FSM model can be established and algorithmic mappings can be developed in order to obtain automated functional size for a selected FSM method. Unified Modeling Language, providing a rich suite of specification elements, is among the promising languages for automated FSM.

Effort Estimation. Effort estimation is a significant practical problem in software engineering. The studies that explore the relationship between size and development effort mostly take total functional size as the primary input; however typed functionality constructs open a new horizon for further exploration of effort-size relationship[146]. 4FSM meta-model Functional Message Capability (FMC) constructs require the identification of the sender and recipient functional user roles. That is, functional user role are the determinants of FMC types and 4FSM facilitates the quantification of the interaction between software and functional user roles. The exploration of effort-functional size relationship based on the FMC types grouped with respect to their functional user roles is an interesting direction for future work.

REFERENCES

[1] A. Abran, *Software metrics and software metrology*. Wiley-IEEE Computer Society Press, 2010.

[2] N. Condori-Fernández and O. Pastor, "Verifying the construction of a software model from a requirements model," in *IX Workshop on Requirements Engineering (WER 2006), Río de Janeiro, Brazil*, 2006.

[3] C. Gencel and O. Demirors, "Functional size measurement revisited," *ACM Transactions on Software Engineering and Methodology*, vol. 17, no. 3, p. 15, 2008.

[4] E. Lamma, P. Mello, and F. Riguzzi, "A system for measuring function points from an er–dfd specification," *The Computer Journal*, vol. 47, no. 3, pp. 358–372, 2004.

[5] C. Gunter, E. Gunter, M. Jackson, and P. Zave, "A reference model for requirements and specifications," *IEEE Software*, vol. 17, no. 3, pp. 37–43, 2000.

[6] R. Wieringa, *Design methods for reactive systems: Yourdon, statemate, and the UML*. Morgan Kaufmann, 2003.

[7] C. Symons, "Come back function point analysis (modernised) - all is forgiven!," in *4th European Conference on Software Measurement and ICT Control (FESMA-DASMA 2001)*, (Germany), pp. 413–426, 2001.

[8] COSMICON, "The cosmic functional size measurement method, v.3.0.1,measurement manual," tech. rep., 2009.

[9] B. Boehm, *Software engineering economics.* Prentice-Hall, 1981.

[10] C. Ebert, R. Dumke, M. Bundschuh, and A. Schmietendorf, *Best Practices in Software Measurement: How to use metrics to improve project and process performance*. Springer-Verlag New York Incorporated, 2005.

[11] N. E. Fenton, *Software Metrics: A Rigorous Approach*. London: Chapman & Hall, Ltd, 1991.

[12] B. Kitchenham, S. Pfleeger, and N. Fenton, "Towards a framework for software measurement validation," *IEEE Transactions on Software Engineering*, vol. 21, no. 12, pp. 929–944, 1995.

[13] A. Albrecht, "Measuring application development productivity," in *Joint SHARE/GUIDE/IBM Application Development Symposium*, vol. 10, pp. 83–92, 1979.

[14] A. J. Albrecht and J. E. Gaffney, "Software function, source lines of code, and development effort prediction: A software science validation," *IEEE Transactions on Software Engineering*, vol. 9, no. 6, pp. 639–648, 1983.

[15] C. Kemerer, "An empirical validation of software cost estimation models," *Communications of the ACM*, vol. 30, no. 5, pp. 416–429, 1987.

[16] J. Matson, B. Barrett, and J. Mellichamp, "Software development cost estimation using function points," *IEEE Transactions on Software Engineering*, vol. 20, no. 4, pp. 275–287, 1994.

[17] D. Garmus, D. Herron, and D. Herron, *Function point analysis: measurement practices for successful software projects*. Addison-Wesley, 2001.

[18] B. Özkan, O. Türetken, and O. Demirörs, "Software functional size: For cost estimation and more," in *EuroSPI*, pp. 59–69, 2008.

[19] S. Furey, "Why we should use function points [software metrics]," *IEEE Software*, vol. 14, no. 2, pp. 28–30, 1997.

[20] E. E. Rudolph, "Productivity in computer application development," tech. rep., Department of Management Studies,University of Auckland, Auckland, New Zealand, 1983.

[21] G. C. Low and D. R. Jeffery, "Function points in the estimation and evaluation of the software process," *IEEE Transactions on Software Engineering*, vol. 16, pp. 64–71, Jan 1990. Cd913 Times Cited:56 Cited References Count:18.

[22] C. F. Kemerer and B. S. Porter, "Improving the reliability of function point measurement: An empirical study," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, pp. 1011–1024, 1992.

[23] C. Kemerer, "Reliability of function points measurement: a field experiment," *Communications of the ACM*, vol. 36, no. 2, pp. 85–97, 1993.

[24] E. Ungan, O. Demirörs, Ö. Ö. Top, and B. Özkan, "An experimental study on the reliability of cosmic measurement results," in *IWSM/Mensura*, pp. 321–336, 2009.

[25] O. O. Top and O. Demirors, "Assessment of the reliability of cosmic fsm results," Tech. Rep. METU/II-TR-2009-17, Middle East Technical University, Informatics Institute., 2009.

[26] O. Turetken, O. Demirors, C. Gencel, O. Ozcan Top, and B. Ozkan, "The effect of entity generalization on software functional sizing: A case study," in *Proceedings of the 9th international conference on Product-Focused Software Process Improvement*, PROFES '08, (Berlin, Heidelberg), pp. 105–116, Springer-Verlag, 2008.

[27] O. Türetken, Ö. Ö. Top, B. Özkan, and O. Demirörs, "The impact of individual assumptions on functional size measurement," in *IWSM/Metrikon/Mensura*, pp. 155–169, 2008.

[28] M. Lother and R. Dumke, "Points metrics-comparison and analysis," in *11th International Workshop on Software Measurement* (R. Dumke and A. Abran, eds.), (Montreal), pp. 228–267, Shaker, 2001.

[29] J. Cuadrado-Gallego, L. Buglione, M. FernÃ¡ndez de Sevilla, P. RodrÃguez-Soria, and M. J. Dominguez, "Horizontal dispersion of software functional size with ifpug and cosmic units," *Advances in Engineering Software*, vol. 41, no. 2, pp. 262 – 269, 2010.

[30] M. Halstead, *Elements of Software Science (Operating and programming systems series)*. Elsevier Science Inc., 1977.

[31] V. Shen, S. Conte, and H. Dunsmore, "Software science revisited: A critical analysis of the theory and its empirical support," *IEEE Transactions on Software Engineering*, no. 2, pp. 155–165, 1983.

[32] R. Glass, *Facts and fallacies of software engineering*. Addison-Wesley Professional, 2003.

[33] C. Jones, *Assessment and control of software risks*. Yourdon Press, 1994.

[34] M. Shepperd, "An evaluation of software product metrics," *Information and Software Technology*, vol. 30, no. 3, pp. 177–188, 1988.

[35] T. DeMarco, *Controlling Software Projects: Management, Measurement, and Estimates*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1986.

[36] C. Jones, "A short history of function points and feature points.," tech. rep., 1986.

[37] C. Symons, *Software sizing and estimating: Mk II FPA (function point analysis)*. John Wiley & Sons, Inc., 1991.

[38] NESMA, "Fpa counting practices manual (cpm 2.0)," 1996.

[39] D. Reifer, "Asset-r: A function point sizing tool for scientific and real-time systems," *Journal of Systems and Software*, vol. 11, no. 3, pp. 159–171, 1990.

[40] S. Whitmire, "3d function points: Scientific and real-time extensions to function points," in *Pacific Northwest Software Quality Conference*, 1992.

[41] R. Banker, R. Kauffman, C. Wright, and D. Zweig, "Automating output size and reuse metrics in a repository-based computer-aided software engineering (case) environment," *IEEE Transactions on Software Engineering*, vol. 20, no. 3, pp. 169–187, 1994.

[42] J. Matson, B. Barrett, and J. Mellichamp, "Software development cost estimation using function points," *IEEE Transactions on Software Engineering*, vol. 20, no. 4, pp. 275–287, 1994.

[43] D. St-Pierre, M. Maya, A. Abran, J. Desharnais, and P. Bourque, "Full function points: Counting practices manual," tech. rep., Software Engineering Management Research Laboratory and Software Engineering Laboratory in Applied Metrics, 1997.

[44] R. Meli, "Early and extended function point: a new method for function points estimation," in *IFPUG Fall Conference*, pp. 15–19, 1997.

[45] G. Caldiera, G. Antoniol, R. Fiutem, and C. Lokan, "Definition and experimental evaluation of function points for object-oriented systems," in *Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International*, pp. 167–178, IEEE, 1998.

[46] G. Teologlou, "Measuring oo software with predictive object points," in *10th Conference on European Software Control and Metrics*, pp. 90–423, 1999.

[47] A. Abran, J. Desharnais, S. Oligny, D. St-Pierre, and C. Symons, “Cosmic-ffp measurement manual, version 2.1,” tech. rep., The Common Software Measurement International Consortium, 2001.

[48] D. J. Reifer, “Web development: Estimating quick-to-market software,” *IEEE Software*, vol. 17, no. 6, pp. 57–64, 2000.

[49] M. Conte, T. Lorio, R. Meli, and L. Santillo, “E&q: An early & quick approach to functional size measurement methods,” in *Proceedings of the 1st Software Metrics European Forum (SMEF 2004), January*, pp. 28–30, 2004.

[50] J. Kammelar, “A sizing approach for oo-environments,” in *Proceedings of the 4th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, 2000.

[51] S. Abrahão, G. Poels, and O. Pastor, “A functional size measurement method for object-oriented conceptual schemas: design and evaluation issues,” *Software and Systems Modeling*, vol. 5, no. 1, pp. 48–71, 2006.

[52] P. Forselius, “Fsm working group: Fisma functional size measurement method v. 1.1.,” tech. rep., Finnish Software Measurement Association (FiSMA), 2004.

[53] ISO/IEC, “Iso/iec 19761:2011 . software engineering – cosmic: a functional size measurement method,” 2011.

[54] ISO/IEC, “Iso/iec 20926:2009. software and systems engineering – software measurement – ifpug functional size measurement method 2009,” 2009.

[55] ISO/IEC, “Iso/iec 20968:2002. software engineering – mk ii function point analysis – counting practices manual,” 2002.

[56] ISO/IEC, “Iso/iec 24570:2005 software engineering – nesma functional size measurement method version 2.1 – definitions and counting guidelines for the application of function point analysis,” 2005.

[57] ISO/IEC, “Iso/iec 29881:2010. information technology – systems and software engineering – fisma 1.1 functional size measurement method,” 2010.

[58] ISO/IEC, “Iso/iec 14143-1:1998. information technology, software measurement, functional size measurement,part 1: Definition of concepts.,” 1998.

[59] ISO/IEC, “Iso/iec 14143-2:2002. information technology, software measurement, functional size measurement,part 2: Conformity evaluation of software size measurement methods to iso/iec-14143-1:1998.,” 2002.

[60] ISO/IEC, “Iso/iec 14143-3:2003. information technology, software measurement, functional size measurement,part 3: Verification of functional size measurement methods.,” 2003.

[61] ISO/IEC, “Iso/iec 14143-4:2002. information technology, software measurement, functional size measurement,part 4: Reference model,” 2002.

[62] ISO/IEC, “Iso/iec 14143-5:2004. information technology, software measurement, functional size measurement, part 5: Determination of functional domains for use with functional size measurement,” 2004.

[63] ISO/IEC, "Iso/iec 14143-6:2006. information technology, software measurement, functional size measurement, part 6: Guide for use of iso/iec 14143 series and related international standards," 2006.

[64] ISBSG, "International software benchmarking standards group dataset." www.isbsg.com, 2011.

[65] IFPUG. www.ifpug.org.

[66] COSMICON. http://www.cosmicon.com/.

[67] NESMA. http://www.nesma.nl.

[68] A. Abran and J. Jacquet, "From software metrics to software measurement methods: A process model," in *Proceedings of the Third International Symposium and Forum on Software Engineering Standards (ISESS97)*, 1997.

[69] T. Fetcke, "A generalized structure for function point analysis," in *International Workshop on Software Measurement*, pp. 143–153, 1999.

[70] S. H. Kan, *Metrics and Models in Software Quality Engineering*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 2002.

[71] B. Kitchenham, "The problem with function points," *IEEE Software*, vol. 14, no. 2, pp. 29–31, 1997.

[72] G. Yilmaz, "The effect of the quality of software requirements document on the functional size measurement," in *International Conference on Software Metrics and Estimating*, 2011.

[73] COSMICON, "Guideline for assuring the accuracy of measurements v1.0," tech. rep., 2011.

[74] E. Codd, "A relational model of data for large shared data banks," *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, 1970.

[75] COSMICON, "Guideline for sizing business application software v1.1," tech. rep., 2008.

[76] B. Potter, D. Till, and J. Sinclair, *An introduction to formal specification and Z*. Prentice Hall PTR, 1996.

[77] C. Jones, *Systematic software development using VDM*, vol. 2. Prentice Hall, 1990.

[78] J. Abrial, *Modeling in Event-B: system and software engineering*. Cambridge Univ Pr, 2010.

[79] C. A. R. Hoare, *Communicating Sequential Processes*. Prentice-Hall, 1985.

[80] R. Milner, *Communicating and mobile systems: the &pgr;-calculus*. New York, NY, USA: Cambridge University Press, 1999.

[81] J. Carlson and B. Lisper, "An event detection algebra for reactive systems," in *Proceedings of the 4th ACM international conference on Embedded software*, pp. 147–154, ACM, 2004.

[82] A. Hinze and A. Voisard, “Eva: An event algebra supporting adaptivity and collaboration in event-based systems,” tech. rep., Citeseer, 2009.

[83] H. Diab, *Formalisation et automatisation de la mesure des points de fonction*. PhD thesis, Department of Informatics FacultÃ© Des Sciences, UniversitÃ© de Sherbrooke, Quebec, Canada, 2003.

[84] V. Bevo, *Analyse et formalisation ontologiques des mesures associées aux méthodes de mesure de la taille fonctionnelle des logiciels: de nouvelles perspectives pour la mesure*. PhD thesis, Ph. D. thesis in Cognitive Infornatics, UQAM, 2005.

[85] H. Diab, M. Frappier, and R. St Denis, “Formalizing cosmic-ffp using room,” in *Computer Systems and Applications, ACS/IEEE International Conference on. 2001*, pp. 312–318, IEEE, 2001.

[86] H. Diab, M. Frappier, and R. St-Denis, “A formal definition of function points for automated measurement of b specifications,” *Formal Methods and Software Engineering*, pp. 483–494, 2002.

[87] M. Heričko, I. Rozman, and A. Živkovič, “A formal representation of functional size measurement methods,” *Journal of Systems and Software*, vol. 79, no. 9, pp. 1341–1358, 2006.

[88] P. Efe, O. Demirors, and C. Gencel, “A unified model for functional size measurement methods,” in *Proceedings 16th Int Workshop Software Measurement*, pp. 343–358, 2006.

[89] O. Demirors and C. Gencel, “Conceptual association of functional size measurement methods,” *IEEE Software*, vol. 26, no. 3, pp. 71–78, 2009.

[90] R. Rask, P. Laamanen, and K. Lyyttinen, “Simulation and comparison of albrecht’s function point and demarco’s function bang metrics in a case environment,” *IEEE Transactions on Software Engineering*, vol. 19, no. 7, pp. 661–671, 1993.

[91] H. B. Diab, F. Koukane, M. Frappier, and R. St.-Denis, “mcrose: automated measurement of cosmic-ffp for rational rose realtime,” *Information and Software Technology*, vol. 47, no. 3, pp. 151–166, 2005.

[92] A. April, E. Merlo, and A. Abran, “A reverse engineering approach to evaluate function point rules,” in *Reverse Engineering, 1997. Proceedings of the Fourth Working Conference on*, pp. 236–245, IEEE, 1997.

[93] T. Miyawaki, J. Iijima, and S. Ho, “Measuring function points from vdm-sl specifications,” in *Service Systems and Service Management, 2008 International Conference on*, pp. 1–6, IEEE, 2008.

[94] T. Fetcke, “Generalized representation for functional size measurement methods,” in *International Workshop on Software Measurement*, pp. 28–29, 2001.

[95] E. V. B. Wandji, G. Lévesque, and J. G. Meunier, “Toward an ontological formalisation for a software functional size measurement method’s application process: The fpa case,” in *RIVF*, pp. 113–118, 2004.

[96] V. Bévo, G. Lévesque, and J. Meunier, "Toward an ontological formalization for a software functional size measurement method's application process: The cosmic ffp case," in *Proceedings of the 13th International Workshop on Software Measurement (IWSM 2003)*, pp. 23–25, 2003.

[97] M. Frappier, "An overview of formal specification languages and their adequacy for formalizing the definition of function points," tech. rep., Département de mathématiques et d'informatique Faculté des sciences Université de Sherbrooke, 1999.

[98] F. Gramantieri, E. Lamma, P. Mello, and F. Riguzzi, "A system for measuring function points from specifications," Tech. Rep. DEIS-LIA-97-006, Universita de Bologna, 1997.

[99] A. Abran and K. Paton, "A formal notation for the rules of function point analysis," tech. rep., Software Engineering Measurement Research Laboratory, Université du Quebec a Montreal (UQAM) Canada, 1995.

[100] R. Rask, "Algorithms for counting unadjusted function points from dataflow diagrams," Tech. Rep. A-1991-1, University of Joensuu, Department of Computer Science, 1991.

[101] IFPUG, "Ifpug function point counting practices: Case studies, case study 2, release 2.0,," tech. rep., 2001.

[102] T. Uemura, S. Kusumoto, and K. Inoue, "Function point measurement tool for uml design specification," in *Software Metrics Symposium, 1999. Proceedings. Sixth International*, pp. 62–69, IEEE, 1999.

[103] T. Fetcke, A. Abran, and T. Nguyen, "Mapping the oo-jacobson approach into function point analysis," in *Technology of Object-Oriented Languages and Systems*, pp. 192–202, IEEE, 1997.

[104] B. Marín, O. Pastor, and G. Giachetti, "Automating the measurement of functional size of conceptual models in an mda environment," in *PROFES*, pp. 215–229, 2008.

[105] N. Condori-Fernández, S. Abrahão, and O. Pastor, "On the estimation of the functional size of software from requirements specifications," *Journal of Computer Science and Technology*, vol. 22, no. 3, pp. 358–370, 2007.

[106] B. Marín, G. Giachetti, and O. Pastor, "Measurement of functional size in conceptual models: A survey of measurement procedures based on cosmic," *Software Process and Product Measurement*, pp. 170–183, 2008.

[107] F. Xia, "On the danger of developing measures without clarifying concepts [software measurement]," in *Software Engineering Conference, 1998. Proceedings. 1998 Asia Pacific*, pp. 86–92, IEEE, 1998.

[108] ISO/IEC/IEEE, "Iso/iec/ieee 24765:2010 systems and software engineering – vocabulary." International Standards Organization & International Electrotechnical Commission, 2010.

[109] M. Jackson, *Problem Frames: Analysing & Structuring Software Development Problems*. Addison-Wesley Professional, Nov. 2000.

[110] P. Zave and M. Jackson, "Four dark corners of requirements engineering," *ACM Transactions on Software Engineering and Methodology*, vol. 6, pp. 1–30, Jan. 1997.

[111] M. Jackson and P. Zave, "Deriving specifications from requirements: an example," in *Proceedings of the 17th international conference on Software engineering*, ICSE '95, (New York, NY, USA), pp. 15–24, ACM, 1995.

[112] I. K. Bray, *An Introduction to Requirements Engineering*. Addison Wesley, 2002.

[113] ComputerDictionary. http://foldoc.org/functionality, 2011.

[114] ISO/IEC, "ISO/IEC 9126-1:2001 - Software engineering – Product quality – Part 1: Quality model," 2001.

[115] B. Chandrasekaran and J. Josephson, "Function in device representation," *Engineering with computers*, vol. 16, no. 3, pp. 162–177, 2000.

[116] G. Roman, "A taxonomy of current issues in requirements engineering," *Computer*, vol. 18, no. 4, pp. 14–23, 1985.

[117] M. Glinz, "Rethinking the notion of non-functional requirements," in *Proc. Third World Congress for Software Quality*, vol. 2, pp. 55–64, 2005.

[118] M. Glinz, "On non-functional requirements," in *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International*, pp. 21–26, Ieee, 2007.

[119] M. Kassab, O. Ormandjieva, M. Daneva, and A. Abran, "Non-functional requirements size measurement method (nfsm) with cosmic-ffp," in *IWSM/Mensura*, pp. 168–182, 2007.

[120] D. Harel and A. Pnueli, *On the development of reactive systems*. Weizmann Institute of Science, Department of Computer Science, 1985.

[121] R. Wieringa, "The declarative problem frame: Designing systems that create and use norms," in *Proceedings of the 10th International Workshop on Software Specification and Design*, p. 75, IEEE Computer Society, 2000.

[122] R. Elmasri and S. Navathe, *Fundamentals of database systems*, vol. 2. Pearson Education India, 2008.

[123] P. Atzeni and V. De Antonellis, *Relational database theory*. Benjamin-Cummings Publishing Co., Inc., 1993.

[124] P. Chen, "The entity - relationship model - toward a unified view of data," *ACM Transactions on Database Systems (TODS)*, vol. 1, no. 1, pp. 9–36, 1976.

[125] A. Makinouchi, "A consideration on normal form of not-necessarily-normalized relation in the relational data model," in *Proc. Intl. Conf. on Very Large Databases*, Citeseer, 1977.

[126] G. Jaeschke and H. Schek, "Remarks on the algebra of non first normal form relations," in *Proceedings of the 1st ACM SIGACT-SIGMOD symposium on Principles of database systems*, pp. 124–138, ACM, 1982.

[127] G. Garani, "A generalised relational data model," *International Journal of Electrical and Electronics Engineering*, vol. 4, no. 1, 2010.

[128] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of databases*, vol. 8. Addison-Wesley, 1995.

[129] R. Hull, "Four views of complex objects: A sophisticate's introduction," in *Nested Relations and Complex Objects in Databases*, vol. 361 of *Lecture Notes in Computer Science*, pp. 85–116, Springer Berlin Heidelberg, 1989.

[130] C. Lecluse, P. Richard, and F. Velez, "O2, an object-oriented data model," *ACM SIGMOD Record*, vol. 17, no. 3, pp. 424–433, 1988.

[131] M. Atkinson, F. Bancilhon, D. DeWitt, K. Dittrich, D. Maier, and S. Zdonik, "The object-oriented database system manifesto," in *Proceedings of the First International Conference on Deductive and Object-Oriented Databases*, vol. 57, 1989.

[132] R. K. Yin, *Case study research: design and methods*. Thousand Oaks: Sage, 1994.

[133] P. Runeson, M. HÃ¶st, A. Rainer, and B. Regnell, *Case Study Research in Software Engineering*. John Wiley & Sons, Inc., 2012.

[134] M. Myers, "Qualitative research in information systems," *MIS Quarterly*, p. 241, 1997.

[135] I. Benbasat, D. Goldstein, and M. Mead, "The case research strategy in studies of information systems," *MIS Quarterly*, pp. 369–386, 1987.

[136] B. Flyvbjerg, "Five misunderstandings about case-study research," *Qualitative Inquiry*, pp. 219–245, 2006.

[137] B. Kitchenham, L. Pickard, and S. Pfleeger, "Case studies for method and tool evaluation," *IEEE Software*, vol. 12, no. 4, pp. 52–62, 1995.

[138] SMRG, "Bütünleşik yazılım kestirim ve araç seti (eeeag proje 107e010)," tech. rep., The Scientific and Technological Research Council of Turkey, 2010.

[139] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *Management Information Systems Quarterly*, vol. 28, no. 1, pp. 75–106, 2004.

[140] B. Kitchenham, S. Linkman, and D. Law, "Desmet: a methodology for evaluating software engineering methods and tools," *Computing & Control Engineering Journal*, vol. 8, no. 3, pp. 120–126, 1997.

[141] COSMICON, "Cosmic group case study:rice cooker,re-organized version nov. 30, 2010," tech. rep., 2010.

[142] COSMICON, "Guideline for sizing real-time software v1.0," tech. rep., 2012.

[143] IFPUG, "Ifpug: case studies 1-3.," tech. rep., The International Function Point Users Group (IFPUG).

[144] G. Yilmaz, "An automated defect detection approach for cosmic functional size measurement method," Master's thesis, Middle East Technical University, Informatics Institute, Turkey, 2012.

[145] M. Salmanoglu, "An error prevention model for cosmic functional size measurement method," Master's thesis, Information Systems Department, Middle East Technical University, 2012.

[146] C. Gencel and L. Buglione, "Do different functionality types affect the relationship between software functional size and effort," in *Proceedings of the Intern. Conf. on Software Process and Product Measurement (IWSM-MENSURA 2007)*, pp. 5–8, 2007.

APPENDIX A

Appendix A: Measurement Report Template for COSMIC FSM

FUR	FP#	FP Name	OOI	DM Type	Data Group Name	Notes

APPENDIX B

Appendix B: Case Study 1-Measurement Results

Table B.1: Measurement Results for Project KA

FUR	FP#	FP Name	DM Type	Data Group Desc.
1	1	Add Vehicle Brand	E	Vehicle Brand Info
			W	Vehicle Brand Info
			X	Error/Confirmation
1	2	List Vehicle Brands	E	Retrive Vehicle Brands request
			R	Vehicle Brands Info
			X	Vehicle Brands Info
1	3	Retrieve Vehicle Brand	E	Retrive Vehicle Brand Detail request
			R	Vehicle Brand Detail Info
			X	Vehicle Brand Detail Info
1	4	Update Vehicle Brand	E	Vehicle Brand Info
			W	Vehicle Brand Info
			X	Error/Confirmation
1	5	Query Vehicle Brands	E	Vehicle Brand Query Request
			R	Vehicle Brand Info
			X	Vehicle Brand Info
2	6	Add Vehicle Type	E	Vehicle Type Info
			W	Vehicle Type Info
			X	Error/Confirmation
2	7	List Vehicle Types	E	Retrive Vehicle Types request
			R	Vehicle Types Info
			X	Vehicle Types Info
2	8	Retrieve Vehicle Type	E	Retrive Vehicle Type Detail request

			R	Vehicle Type Detail Info
			X	Vehicle Type Detail Info
2	9	Update Vehicle Type	E	Vehicle Type Info
			W	Vehicle Type Info
			X	Error/Confirmation
2	10	Query Vehicle Types	E	Vehicle Type Query Request
			R	Vehicle Type Info
			X	Vehicle Type Info
3	11	Add Vehicle Model	E	Vehicle Model Info
			R	Vehicle Brand Info (Dropdown list)
			X	Vehicle Brand Info (Dropdown list)
			W	Vehicle Model Info
			X	Error/Confirmation
3	12	List Vehicle Models	E	Retrive Vehicle Models request
			R	Vehicle Models Info
			X	Vehicle Models Info
3	13	Retrieve Vehicle Model	E	Retrive Vehicle Model Detail request
			R	Vehicle Brand Detail Info
			R	Vehicle Model Detail Info
			X	Vehicle Brand Detail Info
			X	Vehicle Model Detail Info
3	14	Update Vehicle Model	E	Vehicle Model Info
			R	Vehicle Brand Info (Dropdown list)
			X	Vehicle Brand Info (Dropdown list)
			W	Vehicle Model Info
			X	Error/Confirmation
3	15	Query Vehicle Models	E	Vehicle Model Query Request
			R	Vehicle Brand Detail Info
			R	Vehicle Model Detail Info
			X	Vehicle Brand Detail Info
			X	Vehicle Model Detail Info
4	16	Add Vehicle Defect Type	E	Vehicle Defect Type Info
			W	Vehicle Defect Type Info
			X	Error/Confirmation

4	17	List Vehicle Defect Types	E	Retrive Vehicle Defect Types request
			R	Vehicle Defect Types Info
			X	Vehicle Defect Types Info
4	18	Retrieve Vehicle Defect Type	E	Retrive Vehicle Defect Type Detail request
			R	Vehicle Defect Type Detail Info
			X	Vehicle Defect Type Detail Info
4	19	Update Vehicle Defect Type	E	Vehicle Defect Type Info
			W	Vehicle Defect Type Info
			X	Error/Confirmation
4	20	Query Vehicle Defect Types	E	Vehicle Defect Type Query Request
			R	Vehicle Defect Type Info
			X	Vehicle Defect Type Info
5	21	Add Vehicle Rental	E	Vehicle Rental Info
			R	Vehicle Type Info (Dropdown list)
			X	Vehicle Type Info (Dropdown list)
			R	Vehicle Brand Info (Dropdown list)
			X	Vehicle Brand Info (Dropdown list)
			R	Vehicle Model Info (Dropdown list)
			X	Vehicle Model Info (Dropdown list)
			W	Vehicle Rental Info
			X	Error/Confirmation
5	22	List Vehicle Rentals	E	Retrive Vehicle Rentals request
			R	Vehicle Rentals Info
			X	Vehicle Rentals Info
5	23	Retrieve Vehicle Rental	E	Retrive Vehicle Rental Detail request
			R	Vehicle Type Detail Info
			R	Vehicle Model Detail Info
			R	Vehicle Brand Detail Info
			R	Vehicle Rental Detail Info
			X	Vehicle Type Detail Info
			X	Vehicle Model Detail Info
			X	Vehicle Brand Detail Info
			X	Vehicle Rental Detail Info
5	24	Update Vehicle Rental	E	Vehicle Rental Info

			R	Vehicle Type Info (Dropdown list)
			X	Vehicle Type Info (Dropdown list)
			R	Vehicle Brand Info (Dropdown list)
			X	Vehicle Brand Info (Dropdown list)
			R	Vehicle Model Info (Dropdown list)
			X	Vehicle Model Info (Dropdown list)
			W	Vehicle Rental Info
			X	Error/Confirmation
5	25	Query Vehicle Rentals	E	Vehicle Rental Query Request
			R	Vehicle Type Detail Info
			R	Vehicle Model Detail Info
			R	Vehicle Brand Detail Info
			R	Vehicle Rental Detail Info
			X	Vehicle Type Detail Info
			X	Vehicle Model Detail Info
			X	Vehicle Brand Detail Info
			X	Vehicle Rental Detail Info
6	26	Add Vehicle Fuel	E	Vehicle Fuel Info
			R	Vehicle Rental Info (Dropdown list)
			X	Vehicle Rental Info (Dropdown list)
			W	Vehicle Fuel Info
			X	Error/Confirmation
6	27	List Vehicle Fuel	E	Retrive Vehicle Fuel request
			R	Vehicle Rental Detail Info
			R	Vehicle Fuel Detail Info
			X	Vehicle Rental Detail Info
			X	Vehicle Fuel Detail Info
6	28	Retrieve Vehicle Fuel	E	Retrive Vehicle Fuel Detail request
			R	Vehicle Rental Detail Info
			R	Vehicle Fuel Detail Info
			X	Vehicle Rental Detail Info
			X	Vehicle Fuel Detail Info
6	29	Update Vehicle Fuel	E	Vehicle Fuel Info
			R	Vehicle Rental Info (Dropdown list)
			X	Vehicle Rental Info (Dropdown list)
			W	Vehicle Fuel Info
			X	Error/Confirmation
6	30	Query Vehicle Fuel	E	Vehicle Fuel Query Request

			R	Vehicle Rental Detail Info
			R	Vehicle Fuel Detail Info
			X	Vehicle Rental Detail Info
			X	Vehicle Fuel Detail Info
7	31	Add Rented Vehicle Defect	E	Rented Vehicle Defect Info
			R	Vehicle Defect Type Info (Dropdown list)
			X	Vehicle Defect Type Info (Dropdown list)
			R	Vehicle Rental Info (Dropdown list)
			X	Vehicle Rental Info (Dropdown list)
			W	Rented Vehicle Defect Info
			X	Error/Confirmation
7	32	List Rented Vehicle Defects	E	Retrive Rented Vehicle Defects request
			R	Vehicle Defect Type Detail Info
			R	Vehicle Rental Detail Info
			R	Rented Vehicle Defect Detail Info
			X	Vehicle Defect Type Detail Info
			X	Vehicle Rental Detail Info
			X	Rented Vehicle Defect Detail Info
7	33	Retrieve Rented Vehicle Defect	E	Retrive Rented Vehicle Defect Detail request
			R	Vehicle Defect Type Detail Info
			R	Vehicle Rental Detail Info
			R	Rented Vehicle Defect Detail Info
			X	Vehicle Defect Type Detail Info
			X	Vehicle Rental Detail Info
			X	Rented Vehicle Defect Detail Info
7	34	Update Rented Vehicle Defect	E	Rented Vehicle Defect Info
			R	Vehicle Defect Type Info (Dropdown list)
			X	Vehicle Defect Type Info (Dropdown list)
			R	Vehicle Rental Info (Dropdown list)
			X	Vehicle Rental Info (Dropdown list)
			W	Rented Vehicle Defect Info
			X	Error/Confirmation
7	35	Query Rented Vehicle Defects	E	Rented Vehicle Defect Query Request
			R	Vehicle Defect Type Detail Info
			R	Vehicle Rental Detail Info

			R	Rented Vehicle Defect Detail Info
			X	Vehicle Defect Type Detail Info
			X	Vehicle Rental Detail Info
			X	Rented Vehicle Defect Detail Info
8	36	Add Rented Vehicle Daily Wage	E	Rented Vehicle Daily Wage Info
			R	Vehicle Rental Info (Dropdown list)
			X	Vehicle Rental Info (Dropdown list)
			W	Rented Vehicle Daily Wage Info
			X	Error/Confirmation
8	37	List Rented Vehicle Daily Wage	E	Retrive Rented Vehicle Daily Wage request
			R	Vehicle Rental Detail Info
			R	Rented Vehicle Daily Wage Detail Info
			X	Vehicle Rental Detail Info
			X	Rented Vehicle Daily Wage Detail Info
8	38	Retrieve Rented Vehicle Daily Wage	E	Retrive Rented Vehicle Daily Wage Detail request
			R	Vehicle Rental Detail Info
			R	Rented Vehicle Daily Wage Detail Info
			X	Vehicle Rental Detail Info
			X	Rented Vehicle Daily Wage Detail Info
8	39	Update Rented Vehicle Daily Wage	E	Rented Vehicle Daily Wage Info
			R	Vehicle Rental Info (Dropdown list)
			X	Vehicle Rental Info (Dropdown list)
			W	Rented Vehicle Daily Wage Info
			X	Error/Confirmation
8	40	Query Rented Vehicle Daily Wage	E	Rented Vehicle Daily Wage Query Request
			R	Vehicle Rental Detail Info
			R	Rented Vehicle Daily Wage Detail Info
			X	Vehicle Rental Detail Info
			X	Rented Vehicle Daily Wage Detail Info
9	41	Add Rented Vehicle Action	E	Rented Vehicle Action Info

			R	Vehicle Rental Info (Dropdown list)
			X	Vehicle Rental Info (Dropdown list)
			W	Rented Vehicle Action Info
			X	Error/Confirmation
9	42	List Rented Vehicle Action	E	Retrive Rented Vehicle Action request
			R	Vehicle Rental Detail Info
			R	Rented Vehicle Action Detail Info
			X	Vehicle Rental Detail Info
			X	Rented Vehicle Action Detail Info
9	43	Retrieve Rented Vehicle Action	E	Retrive Rented Vehicle Action Detail request
			R	Vehicle Rental Detail Info
			R	Rented Vehicle Action Detail Info
			X	Vehicle Rental Detail Info
			X	Rented Vehicle Action Detail Info
9	44	Update Rented Vehicle Action	E	Rented Vehicle Action Info
			R	Vehicle Rental Info (Dropdown list)
			X	Vehicle Rental Info (Dropdown list)
			W	Rented Vehicle Action Info
			X	Error/Confirmation
9	45	Query Rented Vehicle Action	E	Rented Vehicle Action Query Request
			R	Vehicle Rental Detail Info
			R	Rented Vehicle Action Detail Info
			X	Vehicle Rental Detail Info
			X	Rented Vehicle Action Detail Info

Table B.2: Measurement Results for Project AN

FUR	FP#	FP Name	DM Type	Data Group Desc.
1	1	List Subscribers who have deposit dept	E	List Subscribers who have deposit dept
			R	Subscriber Info
			R	Subscriber Type Info
			R	Reading File Info
			R	Period Info
			R	Adress Info
			X	Period Info

			X	Adress Info
			X	Subscriber Info
			X	Subscriber Type Info
			X	Reading File Info
1	2	Retrieve Subscriber Deposit Info	E	Retrieve Subscriber Deposit Info request
			R	Retrieve Subscriber Info
			X	Retrieve Subscriber Info
			R	Retrieve Subscriber Type
			X	Retrieve Subscriber Type
			R	Retrieve Deposit Fee of the Subscriber Type
			X	Retrieve Deposit Fee of the Subscriber Type
			R	Retrieve Subscriber Adress Info
			X	Retrieve Subscriber Adress Info
1	3	Collect Subscriber Deposit	E	Subscriber's deposit info
			W	Subscriber's deposit info
			X	Error/Confirmation
	4	Export Subscribers' Period Consumption Dept File for Bank Integration	E	Report Selection
2		report	R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			R	Dept Info
			X	Dept Info
			R	Subscriber Info
			X	Subscriber Info
			X	Error/Confirmation
2	5	Download Subscribers' Period Consumption Dept File	N/A	Subscribers Period Consumption Dept File Download Request
			N/A	Error/File Download
2	6	Import Subscribers' Bank Payment File	E	Import Request for Subscribers Bank Payment File
			X	Upload of Subscribers Bank Payment File/Error
			W	Subscribers' Bank Payment Info
			X	Error/Confirmation
2	7	Collect Other Income Items	E	Payment Info
			R	Income Items Info (Income Items Dropdown List)
			X	Income Items Info (Income Items Dropdown List)
			R	Bank Info (Bank Dropdown List)
			X	Bank Info (Bank Dropdown List)
			R	Branch Info (Branch Dropdown List)

			X	Branch Info (Branch Dropdown List)
			R	Payment Type (Payment Type Drop-down List)
			X	Payment Type (Payment Type Drop-down List)
			W	Payment Info
			X	Payment Receipt
3	8	Query&Print Bills	R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			R	Reading File (Reading File Drop-down List)
			X	Reading File (Reading File Drop-down List)
			E	Query Parameters (Period, Reading File, Begin&End Dates)
			R	Subscriber Info
			X	Subscriber Info
			R	Subcribers' Water Consumption Dept Detail
			X	Subcribers' Water Consumption Dept Detail
			R	Subscribers' Water Consumption Depts
			X	Subscribers' Water Consumption Depts
			R	Retrieve Subscriber Adress Info
			X	Retrieve Subscriber Adress Info
4	9	Retrieve Subscriber	E	Subscriber Retrieve Request
			R	Materials Info (Materials Dropdown List)
			X	Materials Info (Materials Dropdown List)
			R	Payment Type (Payment Type Drop-down List)
			X	Payment Type (Payment Type Drop-down List)
			R	Subscriber Info
			X	Subscriber Info
			R	Bank Info (Bank Dropdown List)
			X	Bank Info (Bank Dropdown List)
			R	Branch Info (Branch Dropdown List)
			X	Branch Info (Branch Dropdown List)
			R	Retrieve Subscriber Adress Info
			X	Retrieve Subscriber Adress Info
4	10	Collect First Subscription Payment	E	Payment Record Request

			E	Payment Info
			E	First Subscription Info
			E	Subscriber Contract Info
			E	Subscription Material Info
			R	Payment Parameters Info (By Subscriber Type)
			X	Payment Parameters Info (By Subscriber Type)
			W	Payment Info
			W	First Subscription Info
			W	Subscriber Contract Info
			W	Subscription Material Info
			X	Payment Receipt
5	11	Import Reading File Of Subscribers' Water Meters from Text File	E	Import reading file request
			W	Subcribers' Water Consumption Dept Detail
			W	Subscribers' Water Consumption Depts
			X	Error/Confirmation
6	12	Retrieve subscriber #2	E	Subscriber Retrieve Request
			R	Subscriber Info
			X	Subscriber Info
			R	Retrieve Subscriber Adress Info
			X	Retrieve Subscriber Adress Info
			R	Subscriber Depts Info
			X	Subscriber Depts Info
			R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			R	Bank Info (Bank Dropdown List)
			X	Bank Info (Bank Dropdown List)
			R	Branch Info (Branch Dropdown List)
			X	Branch Info (Branch Dropdown List)
6	11	Collect Any Dept Of A Subsciber	E	Collection of selected debts request
			W	Payment Info
			X	Payment Receipt
7	12	List Payment Parameters	E	List Payment Parameters Request
			R	Payment Parameters Info
			X	Payment Parameters Info
7	13	Add Payment Parameters	E	Payment Parameters Info
			W	Payment Parameters Info
			X	Error/Confirmation
7	14	Retrieve Payment Parameters	E	Retrieve Payment Parameters Detail Request
			R	Payment Parameters Info
			X	Payment Parameters Info
7	15	Update Payment Parameters	E	Payment Parameters Info

			W	Payment Parameters Info
			X	Error/Confirmation
8	16	Create an application	E	Application Info
			R	Application Type Info (Application Type Dropdown List)
			X	Application Type Info (Application Type Dropdown List)
			E	Subscriber Retrieve Request
			R	Subscriber Info
			X	Subscriber Info
			R	Retrieve Subscriber Adress Info
			X	Retrieve Subscriber Adress Info
			E	Record an application request
			W	Application Info
			X	Application Print Out
9	17	Query Applications	E	Query Parameters (Begin&End Dates, Application Number, Status, Application type)
			R	Application Type Info
			X	Application Type Info
			R	Subscriber Info
			X	Subscriber Info
			R	Application Info
			X	Application Info
10	18	Retrieve an application	E	Retrieve Application Request
			R	Subscriber Info
			R	Application Type Info
			R	Adress Info
			X	Adress Info
			X	Subscriber Info
			X	Application Type Info
			R	Application Info
			X	Application Info
11	19	Close an application (Manually closing because of cancelation)	E	Close Application Request
			W	Application Info
			X	Error/Confirmation
12	20	List Application Type	E	List Application Type Request
			R	Application Type Info
			X	Application Type Info
12	21	Add Application Type	E	Application Type Info
			W	Application Type Info
			X	Error/Confirmation
12	22	Retrieve Application Type	E	Retrieve Application Type Detail Request

			R	Application Type Info
			X	Application Type Info
12	23	Update Application Type	E	Application Type Info
			W	Application Type Info
			X	Error/Confirmation
13	24	Query Subscriber History	E	Query Parameters (Subscriber Number)
			R	Subscriber Info
			X	Subscriber Info
			R	Period Info
			X	Period Info
			R	Subscriber Water Consumption Detail(Dept Detail) Info
			X	Subscriber Water Consumption Detail(Dept Detail) Info
			R	Subscriber Water Consumption (Dept Info) Info
			X	Subscriber Water Consumption (Dept Info) Info
			R	Payment Info
			X	Payment Info
14	25	Query Debts	E	Query Parameters (Period,Reading File,Begin&End Amount)
			R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			R	Reading File (Reading File Dropdown List)
			X	Reading File (Reading File Dropdown List)
			R	Subscriber Info
			X	Subscriber Info
			R	Subscriber Dept Info
			X	Subscriber Dept Info
15	26	Query Subscriber Contract Information	E	Query Parameters (Begin&End Dates)
			R	Subscriber Type Info
			X	Subscriber Type Info
			R	Subscriber Contract Info (Count of contracts accourding to subscriber types)
			X	Subscriber Contract Info (Count of contracts accourding to subscriber types)
16	27	Query Subscriber Water Consuption Depts In Detail	R	Period Info (Period Dropdown List)

			X	Period Info (Period Dropdown List)
			E	Query Parameters (Period,Reading File,Debt Type,Subscriber Type,Subscriber Number, Begin&End Amount, Begin&End Water Consumption (m3))
			R	Subscriber Info
			R	Subscriber Dept Info
			X	Subscriber Info
			X	Subscriber Dept Info
16	28	Query Subscriber Water Consuption Depts In Summary	R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			E	Query Parameters (Period,Reading File,Debt Type,Subscriber Type,Subscriber Number, Begin&End Amount, Begin&End Water Consumption (m3))
			R	Sum of subscriber Dept Info
			X	Subscribers' Dept in total
17	29	Query Payments in Detail	R	Reading File (Reading File Dropdown List)
			X	Reading File (Reading File Dropdown List)
			R	Subscriber Debt Type (Subscriber Debt Type Dropdown List)
			X	Subscriber Debt Type (Subscriber Debt Type Dropdown List)
			R	Bank Info (Bank Info Dropdown List)
			X	Bank Info (Bank Info Dropdown List)
			R	Branch Info (Branch Info Dropdown List)
			X	Branch Info (Branch Info Dropdown List)
			R	Subscriber Type (Subscriber Type Dropdown List)
			X	Subscriber Type (Subscriber Type Dropdown List)
			R	Period Info
			X	Period Info
			R	Payment Info
			R	Subscriber Info

			E	Query Parameters (Begin&End Dates,Subscriber Number, Subscriber Type,Reading File, Collector, Sub-scriber Dept Type, Payment Type, Bank , Branch)
			X	Subscriber Info
			X	Payment Info
18	30	Query Payments in Summary	R	Reading File (Reading File Drop-down List)
			X	Reading File (Reading File Drop-down List)
			R	Subscriber Debt Type (Subscriber Debt Type Dropdown List)
			X	Subscriber Debt Type (Subscriber Debt Type Dropdown List)
			R	Bank Info (Bank Info Dropdown List)
			X	Bank Info (Bank Info Dropdown List)
			R	Branch Info (Branch Info Dropdown List)
			X	Branch Info (Branch Info Dropdown List)
			R	Subscriber Type (Subscriber Type Dropdown List)
			X	Subscriber Type (Subscriber Type Dropdown List)
			R	Subscriber Info
			X	Payment Info
			R	
			X	
			E	Query Parameters (Begin&End Dates,Subscriber Number, Subscriber Type,Reading File, Collector, Sub-scriber Dept Type, Payment Type, Bank , Branch)
			X	Subscriber Info
			X	Payment Info
19	31	Cancel Payment	E	Cancel payment request
			W	Payment Info
			X	Error/Confirmation
20	32	Retrieve Receipt	E	Retrieve receipt request
			R	Period Info
			X	Period Info
			R	Dept Type Info

			X	Dept Type Info
			R	Subscriber Info
			X	Subscriber Info
			R	Payment Info
			X	Payment Info
21	33	Retrieve Bill	E	Retrieve bill request
			R	Subscriber Info
			X	Subscriber Info
			R	Adress Info
			X	Adress Info
			R	Period Info
			X	Period Info
			R	Subscriber Type Info
			X	Subscriber Type Info
			R	Debt Info
			X	Debt Info
			R	Debt Detail Info
			X	Debt Detail Info
22	34	Query Water Consumption	R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			E	Query Parameters (Date,Period)
			R	Subscriber Type Info
			X	Subscriber Type Info
			R	Subscriber Water Consumption Info
			X	Water Consumption Report accoutd-ing to subscriber types
23	35	Query Payments according to pay-ment types and periods	R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			E	Query Parameters (Date,Period)
			R	Debt Info
			X	Debt Info
			R	Debt Type Info
			X	Debt Type Info
			R	Payment Info
			X	Payment Report according to period and payment types
24	36	Query Water And Waste Water Con-sumption	R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			E	Query Parameters (Date,Period)
			R	Subscriber Type Info
			X	Subscriber Type Info
			X	Subscriber Info
			X	Subscriber Info

			R	Subscriber Water Consumption Info (Debt Info)
			X	Subscriber Water Consumption Info (Debt Info)
			R	Subscriber Waste Water Consumption Info (Debt Info&Debt Detail Info)
			X	Water&Waste Water Consumption Report accoutding to subscriber types

Table B.3: Measurement Results for Project BM

FUR	FP#	FP Name	DM Type	Data Group Desc.
1	1	List Budget Category	E	List Budget Category Request
			R	Budget Category Info
			X	Budget Category Info
1	2	Add Budget Category	E	Budget Category Info
			R	Upper Budget Category Info (Upper Budget Category Dropdown list)
			X	Upper Budget Category Info (Upper Budget Category Dropdown list)
			W	Budget Category Info
			X	Error/Confirmation
1	3	Retrieve Budget Category	E	Retrieve Budget Category detail request
			R	Upper Budget Category Info (Upper Budget Category Dropdown list)
			X	Upper Budget Category Info (Upper Budget Category Dropdown list)
			R	Budget Category Info
			X	Budget Category Info
1	4	Update Budget Category	E	Budget Category Info
			W	Budget Category Info
			X	Error/Confirmation
2	5	List Budget Item	E	List Budget Item Request
			R	Budget Item Info
			X	Budget Item Info
2	6	Add Budget Item	E	Budget Item Info
			R	Upper Budget Item Info (Upper Budget Item Dropdown list)
			X	Upper Budget Item Info (Upper Budget Item Dropdown list)
			R	Upper Budget Category Info (Upper Budget Category Dropdown list)

			X	Upper Budget Category Info (Upper Budget Category Dropdown list)
			R	Account Info
			X	Account Info
			W	Budget Item Info
			X	Error/Confirmation
2	7	Update Budget Item	E	Budget Item Info
			E	Bugdet Item Account Info
			W	Budget Item Info
			W	Bugdet Item Account Info
			X	Error/Confirmation
2	8	Retrieve Budget Item	E	Retrieve Budget Item detail request
			R	Upper Budget Item Info (Upper Budget Item Dropdown list)
			X	Upper Budget Item Info (Upper Budget Item Dropdown list)
			R	Upper Budget Category Info (Upper Budget Category Dropdown list)
			X	Upper Budget Category Info (Upper Budget Category Dropdown list)
			R	Bugdet Item Account Info
			X	Bugdet Item Account Info
			R	Budget Item Info
			X	Budget Item Info
3	9	Add Budget Request Form	R	Unit Info (Unit Dropdown List)
			X	Unit Info (Unit Dropdown List)
			R	Budget Item Info (Budget Item Dropdown List)
			X	Budget Item Info (Budget Item Dropdown List)
			R	Previous Budget Request Form Info (Readonly Budget Request Form Textbox)
			X	Previous Budget Request Form Info (Readonly Budget Request Form Textbox)
			E	Budget Request Form Info
			W	Budget Request Form Info
			X	Error/Confirmation
3	10	Update Budget Request Form	E	Budget Request Form Info
			W	Budget Request Form Info
			X	Error/Confirmation
3	11	Retrieve Budget Request Form	E	Retrieve Budget Request Form detail request
			R	Unit Info (Unit Dropdown List)

			X	Unit Info (Unit Dropdown List)
			R	Budget Item Info (Budget Item Drop-down List)
			X	Budget Item Info (Budget Item Drop-down List)
			R	Previous Budget Request Form Info (Readonly Budget Request Form Textbox)
			X	Previous Budget Request Form Info (Readonly Budget Request Form Textbox)
			R	Budget Request Form Info
			X	Budget Request Form Info
3	12	Query Budget Request Form	E	Query Parameters (Unit, Budget Itemi, Status)
			R	Budget Request Form Info
			X	Budget Request Form (Unit , Budget Item, Budget Request Amount,Status)
3	13	Add Budget Request Form by Excel File	E	Budget Request Form Excel File Info
			W	Budget Request Form Info
			X	Error/Confirmation
3	14	Approve Budget Request Form	E	Budget Request Form Status Info
			W	Budget Request Form Status Info
			X	Error/Confirmation
3	15	Reject Budget Request Form	E	Budget Request Form Status Info
			W	Budget Request Form Status Info
			X	Error/Confirmation
4	16	Add Fund Request Form	R	Unit Info (Unit Dropdown List)
			X	Unit Info (Unit Dropdown List)
			R	Budget Item Info (Budget Item Drop-down List)
			X	Budget Item Info (Budget Item Drop-down List)
			E	Fund Request Form Info
			W	Fund Request Form Info
			X	Error/Confirmation
4	17	Update Fund Request Form	E	Fund Request Form Info
			W	Fund Request Form Info
			X	Error/Confirmation
4	18	Retrieve Fund Request Form	E	Retrieve Fund Request Form detail request
			R	Unit Info (Unit Dropdown List)
			X	Unit Info (Unit Dropdown List)

			R	Budget Item Info (Budget Item Drop-down List)
			X	Budget Item Info (Budget Item Drop-down List)
			R	Fund Request Form Info
			X	Fund Request Form Info
4	19	Query Fund Request Form	E	Query Parameters (Unit, Bugdet Item, Status)
			R	Fund Request Form Info
			X	Fund Request Form (Unit , Bugdet Item, Fund Request Amount,Status)
4	20	Add Fund Request Form by Excel File	E	Fund Request Form Excel File Info
			W	Fund Request Form Info
			X	Error/Confirmation
4	21	Approve Fund Request Form	E	Fund Request Form Status Info
			W	Fund Request Form Status Info
			X	Error/Confirmation
4	22	Reject Fund Request Form	E	Fund Request Form Status Info
			W	Fund Request Form Status Info
			X	Error/Confirmation
5	23	List Period	E	List Period Request
			R	Period Info
			X	Period Info
5	24	Add Period	E	Period Info
			W	Period Info
			X	Error/Confirmation
5	25	Retrieve Period	E	Retrieve Period Detail Request
			R	Period Info
			X	Period Info
5	26	Update Period	E	Period Info
			W	Period Info
			X	Error/Confirmation
6	27	Transfer Period Parameter	E	Period Parameter Info
			W	Period Info
			X	Error/Confirmation
7	28	Query General Budget	R	Unit Info (Unit Dropdown List)
			X	Unit Info (Unit Dropdown List)
			R	Budget Item Info (Budget Item Drop-down List)
			X	Budget Item Info (Budget Item Drop-down List)
			R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)

			E	Query Parameters (Unit,Budget Item, Period,Planned Budget Item Amounts,
			R	Unit Info
			R	Budget Item Info
			R	Fund Info
			X	Unit Info
			X	Spent Money Info (Spent Money is calculated by accounts that are associated in Budget Item creation from accounting system)
			X	Approved Budget Item Amount, Their Approved Funds and Expenditures of Budget Items
8	29	Change Active Period	R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			E	Active Period Info
			W	Active Period Info
			X	Error/Confirmation
9	30	Move Budget Amount BetweenUnits	R	Unit Info (Unit Dropdown List) Once for source and targetUnit list
			X	Unite Info (Unit Dropdown List) Once for source and targetUnit list
			R	Budget Item Info (Budget Item Dropdown List) Once for source and target budget
			X	Budget Item Info (Budget Item Dropdown List) Once for source and target budget
			E	Budget Move Info
			W	Budget Move Info
			X	Error/Confirmation
10	31	Query Budget Request Report	E	Query Parameters (Unit, Period, Begin&End Dates)
			R	Budget Request Info
			X	Budget Request Info
11	32	Query Fund	E	Query Parameters (Unit,Period, Begin&End Dates)
			R	Unit Info (Unit Dropdown List) Once for source and targetUnite
			X	Unit Item Info (Unit Item Dropdown List) Once for source and targetUnit
			R	Period Info (Period Dropdown List)
			X	Period Info (Period Dropdown List)
			R	Unit Info

			R	Budget Info
			R	Fund Info
			X	Unif Info
			X	Bugdet Info
			X	Fund Info
			R	Spent Money Info (Spent Money is calculated by accounts that are associated in Budget Item creation from accounting system)
			X	Budget Items' Expenditure
12	33	Query Budget Amount Movement	E	Query Parameters (Begin&Date Dates)
			R	Unit Info (Unit Dropdown List) Once for source and targetUnite
			X	Unit Info (Unit Item Dropdown List) Once for source and targetUnit
			R	Bugdet Item
			X	Bugdet Item
			R	Unit Budget Item Info
			X	Unit Budget Item Info
13	34	Query Budget Fund's Expenditure	E	Query Parameters (Begin&End Dates)
			R	Unit Info
			R	Budget Info
			R	Spent Money Info (Spent Money is calculated by accounts that are associated in Budget Item creation from accounting system)
			R	Unit Bugdet Info
			X	Unit Bugdet Info
			X	Unit Info
			X	Bugdet Info
			X	Budget Funds' Expenditure Info
		TOTAL	**175**	**CFP**

Table B.4: Measurement Results for Project CN

FUR	FP Name	DM Type	Data Group Desc.
1	fillDataEntryMenu	E	Authorization Info
		R	Menu Items
		X	Menu Items
2	signIn	E	User Info

		R	Authorization Info
		X	Authorization Info
3	CreateChangeRequest	E	Change Request Info
		E	Problem Attachment Info
		E	Solution Attachment Info
		E	Effected Item Info
		W	Change Request Info
		W	Problem Attachment Info
		W	Solution Attachment Info
		W	Effected Item Info
		X	Error/Confirmation
4	listChangeRequestsWithInitiator	E	Initiator Info
		R	Change Request Info
		X	Change Request Info
5	retrieveChangeRequest	E	Change Request Info
		R	Change Request Detail
		R	Problem Attachment Info
		R	Solution Attachment Info
		R	Effected Item Info
		X	Change Request Detail
		X	Problem Attachment Info
		X	Solution Attachment Info
		X	Effected Item Info
6	UpdateChangeRequest	E	Change Request Detail
		E	Problem Attachment Info
		E	Solution Attachment Info
		E	Effected Item Info
		W	Change Request Detail
		W	Problem Attachment Info
		W	Solution Attachment Info
		W	Effected Item Info
		X	Error/Confirmation
7	deleteChangeRequest	E	Selection of Change Request
		W	Change Request Detail
		W	Problem Attachment Info
		W	Solution Attachment Info
		W	Effected Item Info
		X	Error/Confirmation
8	listChangeRequestsWithProject	E	Project ID
		R	Project Info
		R	Change Request Info
		X	Change Request Info
9	retieveChangeRequestfor review	E	Selection of the Change Request
		R	Change Request Detail

		R	Problem Attachment Info
		R	Solution Attachment Info
		R	Effected Item Info
		X	Change Request Detail
		X	Problem Attachment Info
		X	Solution Attachment Info
		X	Effected Item Info
	review	E	Change Request Review Info
		W	Change Request Review Info
		X	Error/Confirmation
10	retrieveReviewedChangeRequest	E	Change Request Review ID
		R	Change Request Review Info
		X	Change Request Review Info
		X	Error/Confirmation
11	UpdateReviewedChangeRequest	E	Change Request Review Info
		W	Change Request Review Info
		X	Error/Confirmation
12	deleteReviewedChangeRequest	E	Selection of Change Request
		R	Change Request Review Info
		X	Change Request Review Info
		W	Change Request Review Info
		X	Error/Confirmation
13	createCCBAgenda	E	Selection of Change Request
		R	Change Request Detail
		X	Change Request Detail
		E	CCB Agenda Info
		W	CCB Agenda Info
		X	Error/Confirmation
14	listCCBWithProject	E	Project Info
		R	CCB Agenda Info
		X	CCB Agenda Info
	Remove	E	Project Info
		R	CCB Agenda Info
		X	CCB Agenda Info
15	listOpenChangeRequest	E	Project Info
		R	Change Request Info
		X	Change Request Info
16	List for impactAnalysis	E	Selection of Project
		R	Project Info
		X	Project Info
		R	Change Request Detail CCB
		X	Change Request Detail CCB
		E	CCB Agenda Info

		R	Change Request Detail CR
		X	Change Request Detail CR
	Create Impact Analysis	E	Change Request Detail
		E	Problem Attachment
		E	Solution Attachment
		E	Effected Item
		E	CR Item
		E	CR Item Attachment
		E	CR Item Effected Docs
		E	CR Item Action Item
		W	Impact Analysis Info
		W	Change Request Detail
		W	Problem Attachment
		W	Solution Attachment
		W	Effected Item
		W	CR Item
		W	CR Item Attachment
		W	CR Item Effected Docs
		W	CR Item Action Item

Table B.5: Measurement Results for Project GH

FUR	FP#	FP Name	DM Type	Data Group Desc.
1	1	Add Guest House Type	E	Guest House Type Info
			W	Guest House Type Info
			X	Error/Confirmation
1	2	List Guest House Types	E	Retrive Guest House Types request
			R	Guest House Types Info
			X	Guest House Types Info
1	3	Retrieve Guest House Type	E	Retrive Guest House Type Detail request
			R	Guest House Type Detail Info
			X	Guest House Type Detail Info
1	4	Update Guest House Type	E	Guest House Type Info
			W	Guest House Type Info
			X	Error/Confirmation
1	5	Query Guest House Types	E	Guest House Type Query Request
			R	Guest House Type Info

			X	Guest House Type Info
2	6	Add Guest House	E	Guest House Info
			R	Guest House Type Info (Dropdown list)
			X	Guest House Type Info (Dropdown list)
			W	Guest House Info
			X	Error/Confirmation
2	7	List Guest Houses	E	Retrive Guest Houses request
			R	Guest Houses Info
			X	Guest Houses Info
2	8	Retrieve Guest House	E	Retrive Guest House Detail request
			R	Guest House Type Detail Info
			R	Guest House Detail Info
			X	Guest House Type Detail Info
			X	Guest House Detail Info
2	9	Update Guest House	E	Guest House Info
			R	Guest House Type Info (Dropdown list)
			X	Guest House Type Info (Dropdown list)
			W	Guest House Info
			X	Error/Confirmation
2	10	Query Guest Houses	E	Guest House Query Request
			R	Guest House Type Detail Info
			R	Guest House Detail Info
			X	Guest House Type Detail Info
			X	Guest House Detail Info
3	11	Add Guest House Room	E	Guest House Room Info
			R	Guest House Info (Dropdown list)
			X	Guest House Info (Dropdown list)
			W	Guest House Room Info
			X	Error/Confirmation
3	12	List Guest House Rooms	E	Retrive Guest House Rooms request
			R	Guest House Info (Dropdown list)
			X	Guest House Info (Dropdown list)
			R	Guest House Rooms Info
			X	Guest House Rooms Info

3	13	Retrieve Guest House Room	E	Retrive Guest House Room Detail request
			R	Guest House Detail Info
			R	Guest House Room Detail Info
			X	Guest House Detail Info
			X	Guest House Room Detail Info
3	14	Update Guest House Room	E	Guest House Room Info
			R	Guest House Info (Dropdown list)
			X	Guest House Info (Dropdown list)
			W	Guest House Room Info
			X	Error/Confirmation
3	15	Query Guest House Rooms	E	Guest House Room Query Request
			R	Guest House Detail Info
			R	Guest House Room Detail Info
			X	Guest House Detail Info
			X	Guest House Room Detail Info
4	16	Add Guest House Stay Information	E	Guest House Stay Info
			R	Guest House Room Info (Dropdown list)
			X	Guest House Room Info (Dropdown list)
			W	Guest House Stay Info
			X	Error/Confirmation
4	17	List Guest House Stay Information	E	Retrive Guest House Stay Information request
			R	Guest House Info
			R	Guest House Room Info
			R	Guest House Stay Information Detail Info
			X	Guest House Info
			X	Guest House Room Info
			X	Guest House Stay Information Detail Info
4	18	Retrieve Guest House Stay Information	E	Retrive Guest House Stay Information Detail request
			R	Guest House Room Info
			R	Guest House Stay Information Detail Info
			X	Guest House Room Info

			X	Guest House Stay Information Detail Info
4	19	Update Guest House Stay Information	E	Guest House Stay Info
			R	Guest House Room Info (Dropdown list)
			X	Guest House Room Info (Dropdown list)
			W	Guest House Stay Info
			X	Error/Confirmation
4	20	Query Guest House Stay Information	E	Guest House Stay Information Query Request
			R	Guest House Info
			R	Guest House Room Info
			R	Guest House Stay Information Detail Info
			X	Guest House Info
			X	Guest House Room Info
			X	Guest House Stay Information Detail Info
5	21	Add Guest House Stay Fare	E	Guest House Stay Fare Info
			R	Guest House Stay Information (Drop-down list)
			X	Guest House Stay Information (Drop-down list)
			W	Guest House Stay Fare Info
			X	Error/Confirmation
5	22	List Guest House Stay Fares	E	Retrive Guest House Stay Fares request
			R	Guest House Room Info
			R	Guest House Stay Information Detail Info
			R	Guest House Info
			R	Guest House Stay Fares Info
			X	Guest House Room Info
			X	Guest House Stay Information Detail Info
			X	Guest House Info
			X	Guest House Stay Fares Info
5	23	Retrieve Guest House Stay Fare	E	Retrive Guest House Stay Fare Detail request

			R	Guest House Stay Information Detail Info
			R	Guest House Stay Fares Info
			X	Guest House Stay Information Detail Info
			X	Guest House Stay Fares Info
5	24	Update Guest House Stay Fare	E	Guest House Stay Fare Info
			R	Guest House Stay Information (Drop-down list)
			X	Guest House Stay Information (Drop-down list)
			W	Guest House Stay Fare Info
			X	Error/Confirmation
5	25	Query Guest House Stay Fares	E	Guest House Stay Fare Query Request
			R	Guest House Room Info
			R	Guest House Stay Information Detail Info
			R	Guest House Info
			R	Guest House Stay Fares Info
			X	Guest House Room Info
			X	Guest House Stay Information Detail Info
			X	Guest House Info
			X	Guest House Stay Fares Info
6	26	Add Guest House Stay Detail Information	E	Guest House Stay Detail Info
			R	Guest House Stay Information (Drop-down list)
			X	Guest House Stay Information (Drop-down list)
			W	Guest House Stay Detail Info
			X	Error/Confirmation
6	27	List Guest House Stay Detail Information	E	Retrive Guest House Stay Detail Information request
			R	Guest House Stay Information Detail Info
			R	Guest House Stay Detail Information Info
			X	Guest House Stay Information Detail Info

			X	Guest House Stay Detail Information Info
6	28	Retrieve Guest House Stay Detail Information	E	Retrive Guest House Stay Detail Information Detail request
			R	Guest House Room Info
			R	Guest House Stay Information Detail Info
			R	Guest House Info
			R	Guest House Stay Detail Information Info
			X	Guest House Room Info
			X	Guest House Stay Information Detail Info
			X	Guest House Info
			X	Guest House Stay Detail Information Info
6	29	Update Guest House Stay Detail Information	E	Guest House Stay Detail Info
			R	Guest House Stay Information (Drop-down list)
			X	Guest House Stay Information (Drop-down list)
			W	Guest House Stay Detail Info
			X	Error/Confirmation
6	30	Query Guest House Stay Detail Information	E	Guest House Stay Detail Information Query Request
			R	Guest House Stay Information Detail Info
			R	Guest House Stay Detail Information Info
			X	Guest House Stay Information Detail Info
			X	Guest House Stay Detail Information Info

Table B.6: Measurement Results for Project HB

FUR	FP#	FP Name	DM Type	Data Group Desc.
1	1	Clientlari Listele	E	Client Listeleme Istegi

			R	Client Bilgisi
			X	Client Bilgisi
1	2	Ayarlari Goster	E	Ayar Getirme Istegi
			R	Ayar Bilgisi
			X	Ayar Bilgisi
1	3	Ayarlari Degistir	E	Ayar Bilgisi
			W	Ayar Bilgisi
			X	Onay/Hata
1	4	Ayarlari Kaydet	E	Ayar Bilgisi
			W	Ayar Bilgisi
			X	Onay/Hata
1	5	Kurallari Goster	E	Kural Listeleme Istegi
			R	Kural Bilgisi
			X	Kural Bilgisi
1	6	Yeni Kural Kaydet	E	Kural Bilgisi
			W	Kural Bilgisi
			X	Onay/Hata
2	7	Borc Hesapla	E	Borc Hesaplama Istegi
			R	Tahakkuk Bilgisi
			R	Thksatir Bilgisi
			R	Thksatirhesaplanan Bilgisi
			R	Thktaksit Bilgisi
			R	odemeplaniana Bilgisi
			R	odemeplani Bilgisi
			R	odemeplanisatir Bilgisi
			R	opshesaplanan Bilgisi
			R	Odeme Bilgisi
			R	Tahsilat Bilgisi
			R	Odemesatirhesaplanan Bilgisi
			R	hesaplananvergiodeme Bilgisi
			R	odemesatirhesaplanan Bilgisi
			R	tahsilatekbilgi Bilgisi
			R	odemeekbilgi Bilgisi
			R	tahsilatdetaylari Bilgisi
			R	tahsilataciklama Bilgisi
			R	odemedetay Bilgisi
			R	odemebankabilgileri Bilgisi
			R	odemeemanetbilgileri Bilgisi
			R	odemeindirimbilgileri Bilgisi
			R	odemesaymanlikbilgileri Bilgisi
			R	cekbilgileri Bilgisi
			R	duzeltme Bilgisi
			R	duzeltmebelgeleri Bilgisi
			R	duzeltmeiade Bilgisi

			R	duzeltmeiadesatir Bilgisi
			R	duzeltmesonuclari Bilgisi
			R	tecil414 Bilgisi
			R	tecil414detay Bilgisi
			R	tecil5335 Bilgisi
			R	tecil5335detay Bilgisi
			R	HACIZVARAKA Bilgisi
			R	HACIZVARAKAEKBLG Bilgisi
			R	HACIZVARAKAZIMMET Bilgisi
			R	HACIZVARAKAISLEMLER Bilgisi
			R	HACIZSATIR Bilgisi
			R	HACIZMALLAR Bilgisi
			R	HACIZBILDIRISI Bilgisi
			R	IHTTEMHACIZ Bilgisi
			R	IHTTEMHACIZSATIR Bilgisi
			R	HACIZBILMENKULMAL Bilgisi
			R	HACIZZIMMETISLEMLERI Bilgisi
			R	HACIZBILGAYRIMENKULMAL Bilgisi
			R	HATALIHACIZZIMMETDUSME Bilgisi
			R	HACIZBILDIRISIVARAKALAR Bilgisi
			R	TKPRELATIONS Bilgisi
			R	SRKZTPC Bilgisi
			R	SRKZTPCEKBLG Bilgisi
			R	SRKZOKDTPCORTAK Bilgisi
			R	SRKZOLAYKAYIT Bilgisi
			R	SRKZOKDEKBLG Bilgisi
			R	SRKZOKDSATIR Bilgisi
			R	SRKZEKBLG Bilgisi
			R	SRKZTAPUEKBILGI Bilgisi
			R	SRKZKABAHATTAKSIT Bilgisi
			R	TAKIP Bilgisi
			R	TKPISLEMLER Bilgisi
			R	TKPSATIRLAR Bilgisi
			R	TKPRELATIONS Bilgisi
			R	TKPILANLISTESI Bilgisi
			R	TKPIDARIBILGILERI Bilgisi
			R	HAPSENTAZYIK Bilgisi
			R	HAPSENTAZYIKISLEM Bilgisi
			R	HAPSENTAZYIKTAKIP Bilgisi
			R	dztTahakkuk Bilgisi
			R	dztThksatir Bilgisi
			R	dztThksatirhesaplanan Bilgisi

			R	dztThktaksit Bilgisi
			R	dztodemeplaniana Bilgisi
			R	dztodemeplani Bilgisi
			R	dztodemeplanisatir Bilgisi
			R	dztopshesaplanan Bilgisi
			R	dztOdeme Bilgisi
			R	dztTahsilat Bilgisi
			R	dztOdemesatirhesaplanan Bilgisi
			R	dzthesaplananvergiodeme Bilgisi
			R	dztodemesatirhesaplanan Bilgisi
			R	dzttahsilatekbilgi Bilgisi
			R	dztodemeekbilgi Bilgisi
			R	dzttahsilatdetaylari Bilgisi
			R	dzttahsilataciklama Bilgisi
			R	dztodemedetay Bilgisi
			R	dztodemebankabilgileri Bilgisi
			R	dztodemeemanetbilgileri Bilgisi
			R	dztodemeindirimbilgileri Bilgisi
			R	dztodemesaymanlikbilgileri Bilgisi
			R	dztcekbilgileri Bilgisi
			W	Hazirborc Bilgisi
			W	Tahsilat Bilgisi
3	8	Degisiklik Sorgula	E	Degisiklik Sorgulama Istegi
			R	Degisiklik Bilgisi
			X	Degisiklik Bilgisi
3	9	Borc Sorgula	E	Borc Kriteri
			R	Hazirborc Bilgisi
			R	Tahsilat Bilgisi
			W	Tahakkuk Bilgisi
			W	Thksatir Bilgisi
			W	Thksatirhesaplanan Bilgisi
			W	Thktaksit Bilgisi
			W	odemeplaniana Bilgisi
			W	odemeplani Bilgisi
			W	odemeplanisatir Bilgisi
			W	opshesaplanan Bilgisi
			W	Odeme Bilgisi
			W	Tahsilat Bilgisi
			W	Odemesatirhesaplanan Bilgisi
			W	hesaplananvergiodeme Bilgisi
			W	odemesatirhesaplanan Bilgisi
			W	tahsilatekbilgi Bilgisi
			W	odemeekbilgi Bilgisi
			W	tahsilatdetaylari Bilgisi

			W	tahsilataciklama Bilgisi
			W	odemedetay Bilgisi
			W	odemebankabilgileri Bilgisi
			W	odemeemanetbilgileri Bilgisi
			W	odemeindirimbilgileri Bilgisi
			W	odemesaymanlikbilgileri Bilgisi
			W	cekbilgileri Bilgisi
			W	duzeltme Bilgisi
			W	duzeltmebelgeleri Bilgisi
			W	duzeltmeiade Bilgisi
			W	duzeltmeiadesatir Bilgisi
			W	duzeltmesonuclari Bilgisi
			W	tecil414 Bilgisi
			W	tecil414detay Bilgisi
			W	tecil5335 Bilgisi
			W	tecil5335detay Bilgisi
			W	HACIZVARAKA Bilgisi
			W	HACIZVARAKAEKBLG Bilgisi
			W	HACIZVARAKAZIMMET Bilgisi
			W	HACIZVARAKAISLEMLER Bilgisi
			W	HACIZSATIR Bilgisi
			W	HACIZMALLAR Bilgisi
			W	HACIZBILDIRISI Bilgisi
			W	IHTTEMHACIZ Bilgisi
			W	IHTTEMHACIZSATIR Bilgisi
			W	HACIZBILMENKULMAL Bilgisi
			W	HACIZZIMMETISLEMLERI Bilgisi
			W	HACIZBILGAYRIMENKULMAL Bilgisi
			W	HATALIHACIZZIMMETDUSME Bilgisi
			W	HACIZBILDIRISIVARAKALAR Bilgisi
			W	TKPRELATIONS Bilgisi
			W	SRKZTPC Bilgisi
			W	SRKZTPCEKBLG Bilgisi
			W	SRKZOKDTPCORTAK Bilgisi
			W	SRKZOLAYKAYIT Bilgisi
			W	SRKZOKDEKBLG Bilgisi
			W	SRKZOKDSATIR Bilgisi
			W	SRKZEKBLG Bilgisi
			W	SRKZTAPUEKBILGI Bilgisi
			W	SRKZKABAHATTAKSIT Bilgisi
			W	TAKIP Bilgisi
			W	TKPISLEMLER Bilgisi

			W	TKPSATIRLAR Bilgisi
			W	TKPRELATIONS Bilgisi
			W	TKPILANLISTESI Bilgisi
			W	TKPIDARIBILGILERI Bilgisi
			W	HAPSENTAZYIK Bilgisi
			W	HAPSENTAZYIKISLEM Bilgisi
			W	HAPSENTAZYIKTAKIP Bilgisi
			W	dztTahakkuk Bilgisi
			W	dztThksatir Bilgisi
			W	dztThksatirhesaplanan Bilgisi
			W	dztThktaksit Bilgisi
			W	dztodemeplaniana Bilgisi
			W	dztodemeplani Bilgisi
			W	dztodemeplanisatir Bilgisi
			W	dztopshesaplanan Bilgisi
			W	dztOdeme Bilgisi
			W	dztTahsilat Bilgisi
			W	dztOdemesatirhesaplanan Bilgisi
			W	dzthesaplananvergiodeme Bilgisi
			W	dztodemesatirhesaplanan Bilgisi
			W	dzttahsilatekbilgi Bilgisi
			W	dztodemeekbilgi Bilgisi
			W	dzttahsilatdetaylari Bilgisi
			W	dzttahsilataciklama Bilgisi
			W	dztodemedetay Bilgisi
			W	dztodemebankabilgileri Bilgisi
			W	dztodemeemanetbilgileri Bilgisi
			W	dztodemeindirimbilgileri Bilgisi
			W	dztodemesaymanlikbilgileri Bilgisi
			W	dztcekbilgileri Bilgisi
			X	Additional OOI for report
3	10	Calisma durumu gUncelle	E	Calisma GUncelleme Istegi
			W	Durum Bilgisi
			X	Durum Bilgisi

Table B.7: Measurement Results for Project MN

FUR	**FP#**	**FP Name**	**DM Type**	**Data Group Desc.**
1	1	Login	E	User info
			R	User info
			X	Error/Confirmation

2	2	Retrieve Menu Bar	E	Login
			R	Menu Info
			R	Kullanici
			R	Rol
			X	Menu Bar
3	3	Add Firm	E	Firm Info
			W	Firm Info
			X	Error/Confirmation
3	4	List Firms	E	Retrive firms request
			R	Firms info
			X	Firms info
3	5	Retrieve Firm	E	Retrive firm detail request
			R	Firm detail info
			X	Firm detail info
3	6	Update Firm	E	Firm info
			W	Firm info
			X	Error/Confirmation
4	7	Add Project Definition	E	Project Definition Info
			W	Project Definition Info
			X	Error/Confirmation
4	8	List Project Definitions	E	List project definition request
			R	Project Definitions info
			X	Project Definitions info
4	9	Retrieve Project Definition	E	Retrive firm detail request
			R	Project Definition detail info
			X	Project Definition detail info
4	10	Update Project Definition	E	Project Definition info
			W	Project Definition info
			X	Error/Confirmation
5	11	Add Project Type	E	Project Type Info
			W	Project Type Info
			X	Error/Confirmation
5	12	List Project Types	E	List project types request
			R	Project Types info

			X	Project Types info
5	13	Retrieve Project Type	E	Retrive firm detail request
			R	Project Type detail info
			X	Project Type detail info
5	14	Update Project Type	E	Project Type info
			W	Project Type info
			X	Error/Confirmation
6	15	Add Corporation	E	Corporation Info
			W	Corporation Info
			X	Error/Confirmation
6	16	List Corporations	E	List corporations request
			R	Corporations info
			X	Corporations info
6	17	Retrieve Corporation	E	Retrive corporation detail request
			R	Corporation detail info
			X	Corporation detail info
6	18	Update Corporation	E	Corporation info
			W	Corporation info
			X	Error/Confirmation
6	19	Add Bank	E	Bank Info
			R	Firm Info (dropdown list)
			X	Firm Info (dropdown list)
			W	Bank info
			X	Error/Confirmation
6	20	List Banks	E	List banks request
			R	Banks info
			X	Banks info
6	21	Retrieve Bank	E	Selection of Bank
			R	Bank Info
			R	Firm Info
			X	Bank Info
			X	Firm Info
6	22	Update Bank	E	Bank Info
			R	Firm Info (dropdown list)

			X	Firm Info (dropdown list)
			W	Bank info
			X	Error/Confirmation
6	23	Add Branch	E	Branch Info
			R	Bank Info (dropdown list)
			X	Bank Info (dropdown list)
			W	Branch info
			X	Error/Confirmation
6	24	List Branchs	E	List banks request
			R	Branchs info
			X	Branchs info
6	25	Retrieve Branch	E	Selection of Branch
			R	Branch Info
			R	Bank Info
			X	Branch Info
			X	Bank Info
6	26	Update Branch	E	Branch Info
			R	Bank Info (dropdown list)
			X	Bank Info (dropdown list)
			W	Branch info
			X	Error/Confirmation
6	27	Add Currency	E	Currency Info
			R	Branch Info (dropdown list)
			X	Branch Info (dropdown list)
			R	Currency info (dropdown list)
			X	Currency info (dropdown list)
			W	Currency info
			X	Error/Confirmation
6	28	List Currencies	E	List currencies request
			R	Currencies info
			X	Currencies info
6	29	Retrieve Currency	E	Selection of Currency
			R	Currency Info
			R	Branch Info
			X	Currency Info
			X	Branch Info

6	30	Add Account Number	E	Account Number Info
			R	Currency Info (dropdown list)
			X	Currency Info (dropdown list)
			R	Corporation Info (dropdown list)
			X	Corporation Info (dropdown list)
			R	Bank Info (drop down list)
			X	Bank Info (drop down list)
			R	Branch Info(drop down list)
			X	Branch Info(drop down list)
			W	Account Number info
			X	Error/Confirmation
6	31	List Account Numbers	E	List account numbers request
			R	Account Numbers info
			X	Account Numbers info
6	32	Retrieve Account Number	E	Selection of Account Number
			R	Account Number Info
			R	Currency Info
			X	Account Number Info
			X	Currency Info
6	33	Update Account Number	E	Account Number Info
			R	Currency Info (dropdown list)
			X	Currency Info (dropdown list)
			W	Account Number info
			X	Error/Confirmation
7	34	Add Letter of Credit	E	Letter of Credit Info
			W	Cash outflow info
			W	Cash inflow info
			X	Error/Confirmation
		deletion	E	
			W	
			W	
			X	
8	35	Add Sale of Foreign Currency	E	Letter of Credit Info
			W	Cash outflow info
			W	Cash inflow info
			X	Error/Confirmation
		Deletion of Foreign Currency	E	
			W	

			W	
			X	
9	36	Add Purchase of Foreign Currency	E	Letter of Credit Info
			W	Cash outflow info
			W	Cash inflow info
			X	Error/Confirmation
		Deletion	E	
			W	
			W	
10			X	
	37	Add Payments Made to Other Corporations	E	Letter of Credit Info
			W	Cash outflow info
			W	Cash inflow info
			X	Error/Confirmation
		deletion	E	
			W	
11			W	
			X	
	38	Add Vault Operation	E	Letter of Credit Info
			W	Cash outflow info
			W	Cash inflow info
12			X	Error/Confirmation
		Deletion	E	
			W	
			W	
			X	
13	39	Add Credit Payment /Usage	E	Letter of Credit Info
			W	Cash outflow info
			W	Cash inflow info
			X	Error/Confirmation
		Deletion	E	
			W	
14			W	
			X	
	40	Add Payment to Foreign Countries	E	Letter of Credit Info
			W	Cash outflow info
			W	Cash inflow info
15			X	Error/Confirmation
		Deletion	E	
			W	

			W	
			X	
16	41	Add Payment to Firms	E	Letter of Credit Info
			W	Cash outflow info
			W	Cash inflow info
			X	Error/Confirmation
		Deletion	E	
			W	
			W	
			X	
	42	Print Cash Outflow	E	Print Cash Outflow
			R	Paragirisi
			R	Paracikisi
			X	Paragirisi
			X	Paracikisi
	43	Query Past Operation	E	Enter Past Operation Query Info
			R	Paragirisi
			R	Paracikisi
			R	Projeaciklama
17			R	Tur
			R	Firma
			R	HesapNo
			R	Banka
			R	Yetkilikisi
			X	Paragirisi
			X	Paracikisi
			X	Projeaciklama
			X	Tur
			X	Firma
			X	HesapNo
			X	Banka
			X	Yetkilikisi
	44	Query Daily Operations	E	Enter Daily Operation Query Info
			R	Paragirisi
			R	Paracikisi
			R	Projeaciklama
18			R	Tur
			R	Firma
			R	HesapNo
			R	Banka
			R	Yetkilikisi
			X	Paragirisi

			X	Paracikisi
			X	Projeaciklama
			X	Tur
			X	Firma
			X	HesapNo
			X	Banka
			X	Yetkilikisi
	45	Query Cash Outflow	E	Enter Cash Outflow Query Info
			R	Paragirisi
			R	Paracikisi
			R	Projeaciklama
			R	Tur
			R	Firma
			R	HesapNo
			R	Banka
			R	Yetkilikisi
			X	Paragirisi
			X	Paracikisi
			X	Projeaciklama
			X	Tur
			X	Firma
			X	HesapNo
			X	Banka
			X	Yetkilikisi

Table B.8: Measurement Results for Project SN

FUR	FP Name	DM Type	Data Group Desc.
1	signIn	E	User Info
		R	Authorization Info
		X	Error/Confirmation
2	listMaterials	R	Project Info(just once for Project Name dropdown lists)
		X	Project Info(just once for Project Name dropdown lists)
		E	Material Info
		R	Material Info
		X	Material Info
3	retrieveMaterial	E	Selected Material Info
		R	Material Detail Info
		X	Material Detail Info

4	createRequisition	E	Requisition Info
		E	Material Info
		E	Project Info
		E	User Info
		W	Requisition Info
		W	Material Info
		W	Project Info
		W	User Info
		X	Error/Confirmation
5	listRequisitionStatus	R	Department Info (just once for Department Name dropdown lists)
		X	Department Info (just once for Department Name dropdown lists)
		E	Requisition Info
		E	Department Info
		E	Personnel Info
		R	Requisition Info
		X	Requisition Info
6	UpdateRequisition	E	Selected Requisition Info
		R	Requisition Info
		R	Material Info
		R	Project Info
		R	User Info
		W	Requisition Info
		W	Material Info
		W	Project Info
		W	User Info
		X	Error/Confirmation
7	listRequisitionApproval	R	Department Info (just once for Department Name dropdown lists)
		X	Department Info (just once for Department Name dropdown lists)
		E	Requisition Info
		E	Department Info
		E	Personnel Info
		R	Requisition Info
		X	Requisition Info
8	approveRequisition	E	Selected Requisition Info
		R	Requisition Info
		R	Material Info
		R	Project Info
		R	User Info
		W	Requisition Info
		X	Error/Confirmation
9	cancelRequisition	E	Selected Requisition Info

		R	Requisition Info
		R	Material Info
		R	Project Info
		R	User Info
		W	Requisition Info
		X	Error/Confirmation
10	getReport	E	Requisition Info
		E	Material Info
		E	Project Info
		E	User Info
		R	Requisition Info
		R	Material Info
		R	Project Info
		R	User Info
		X	Requisition Info
		X	Material Info
		X	Project Info
		X	User Info

Table B.9: Measurement Results for Project TCL

FUR	FP Name	DM Type	Data Group Desc.
1	EVRAK BILGILERI GETIR		
		E	DILEKCE BILGILERI
		R	DILEKCE BILGILERI
		X	DILEKCE BILGILERI
2	KIMLIK BILGILERI GETIR		
		E	MUKELLEF BILGILERI
		R	MUKELLEF BILGILERI
		X	MUKELLEF BILGILERI
		X	MESSAGE [18]
3	ARAC BILGISI / SAHIP BILGISI GETIR		
		E	PLAKA ILE ARAC BILGISI
		R	PLAKA ILE ARAC BILGISI
		R	Sahip Bilgisi
		X	PLAKA ILE ARAC BILGISI
		X	Sahip Bilgisi
4	TECIL ANA BILGILERI TOPLA		

	tek bir FP	E	TECIL ANA BILGILERI
		E	OZEL TAHAKKUK BILGISI
		E	ONAYLAYAN MAKAM
		W	TECIL ANA BILGILERI
		W	OZEL TAHAKKUK BILGISI
5	TECIL EK BILGILERI TOPLA		
		E	TECIL EK BILGILERI
		E	DERHAL ODEME BILGISI
		E	DERHAL ODEME MIKTAR
		E	BELEDIYE BORCU
		W	TECIL EK BILGILERI
		W	DERHAL ODEME BILGISI
6	ORTAK BILGILERI TOPLA		
		E	ORTAK BILGILERI
		R	ORTAK BILGILERI
		W	ORTAK BILGILERI
7	BORCLU TAHAKKUKLARI GETIR		
		E	BORC GIRIS KRITERLERI
		R	BORCLU SATIRLAR
		X	BORCLU TAHAKKUKLAR LISTESI
		X	BORCLU D03 TAHAKKUKLAR LISTESI
8	BORCLU TAHAKKUKLARI BELIRLE		
		E	D03 TAHAKKUK ANA VERGI KODU
		E	D03 TAHAKKUK VADE BILGISI
		X	TECILE ALINACAK D03 TAHAKKUKLARI
		E	TECILE ALINACAK D03 TAHAKKUKLARI
		X	TECILE ALINACAK D03 TAHAKKUK VADELERI
		E	TECILE ALINACAK D03 VADESI
		X	TECILE ALINACAK D03 VADESI
		E	TECILE ALINACAK D03 BORC MIKTARI [1]
		E	TAHAKKUK ANA VERGI KODU
		E	TAHAKKUK VADE BILGISI
		X	TECILE ALINACAK TAHAKKUKLAR
		E	TECILE ALINACAK TAHAKKUKLAR

		X	TECILE ALINACAK TAHAKKUK VADELERI
		E	TECILE ALINACAK TAHAKKUK VADELERI
		E	TECILE ALINACAK TAHAKKUK BORC MIKTARI [3]
		E	DILEKCE BILGILERI
	OOI Read	R	GECIKME ZAMMI ORANINI AL
		R	TPC GECIKME ZAMMI ORANINI AL
		E	TOPLAM D03 VERGI ASLI BORCU
		E	TOPLAM D03 GECIKME ZAMMI BORCU
		E	TOPLAM TAHAKKUK VERGI ASLI BORCU
		E	TOPLAM TAHAKKUK GECIKME ZAMMI BORCU
		X	TOPLAM D03 VERGI ASLI BORCU
		X	TOPLAM D03 GECIKME ZAMMI BORCU
		X	TOPLAM TAHAKKUK VERGI ASLI BORCU
		X	TOPLAM TAHAKKUK GECIKME ZAMMI BORCU
9	ODEME PLANI		
	OLUSTUR		
		E	TAKSIT SAYISI
		E	DILEKCE BILGILERI
		E	ODEME VADE [1]
		E	ODEME VADE MIKTARI [2]
		E	ODEME PLANI TOPLAM MIKTAR
		X	ODEME PLANI TOPLAM MIKTAR
		R	BIR AY SONRASI BILGISI
		R	BIR AY SONRAKI SON IS GUNU BILGISI (BIRAYSONRASI 2)
		R	GECERLI VADE
		X	MESSAGE [10]
		X	MESSAGE [11]
		R	TECIL LIMITLERINI AL
		X	MESSAGE [21]
		X	MESSAGE [20]
		X	MESSAGE [22]
		X	MESSAGE [19]

10	TECIL TAKSITLERI OLUSTUR		
		E	VERGI KODLARI
		R	VERGI KODLARI
		X	MESSAGE [05]
		X	MESSAGE [06]
		X	MESSAGE [02]
		R	TAHAKKUK GECICI DUZELTME BILGILERI
		R	D03 TAHAKKUK GECICI DUZELTME BILGILERI
		X	MESSAGE [03]
		X	MESSAGE [04]
		X	MESSAGE [09]
		X	MESSAGE [12]
		R	TAHAKKUK BILGISI
		X	MESSAGE [15]
		X	MESSAGE [16]
		X	MESSAGE [07]
		X	MESSAGE [08]
		X	MESSAGE [13]
		R	GECERLI VADE
		X	MESSAGE [17]
		X	MESSAGE [14]
		R	TECIL FAIZ ORANI
		X	TECIL FAIZI
		E	TECIL PLANI TOPLAM VERGI ASLI BORCU
		X	TECIL PLANI TOPLAM VERGI ASLI BORCU
		R	TECIL PLANI TOPLAM VERGI ASLI BORCU
		E	TECIL PLANI TOPLAM GECIKME ZAMMI BORCU
		R	TECIL PLANI TOPLAM GECIKME ZAMMI BORCU
		X	TECIL PLANI TOPLAM GECIKME ZAMMI BORCU
		E	TECIL PLANI TOPLAM VERGI ASLI BORCU
		X	TECIL PLANI TOPLAM VERGI ASLI BORCU
		X	TECIL TAKSITLERI
		X	D03 TECIL TAKSITLERI
11	TECILI KAYDET		

		E	KULLANICI BILGILERI
		R	KULLANICI BILGILERI
		E	TERMINAL BILGISI
		R	TERMINAL BILGISI
		E	SERVIS BILGISI
		R	SERVIS BILGISI
		E	TECIL DOSYANO
		R	TECIL DOSYANO
		X	TECIL DOSYANO
		W	TECIL ANA BILGILERI
		W	TECIL EK BILGILERI
		W	ORTAK BILGILERI
		W	TECIL TAKSITLERI
		W	D03 TECIL TAKSITLERI
		W	TECILLI TAHAKKUKLAR
		W	TAHAKKUK
12	YAZDIR		
		E	VERGI DAIRESI BILGISI
		R	VERGI DAIRESI BILGISI
		X	VERGI DAIRESI BILGISI
		X	TECIL ANA BILGILERI
		X	ODEME PLANI
		X	TECIL TAKSITLERI
		X	D03TECI LTAKSITLERI
		E	TOPLAM VERGI ASLI
		X	TOPLAM VERGI ASLI
		E	TOPLAM GECIKME ZAMMI
		X	TOPLAM GECIKME ZAMMI
		E	TOPLAM TECIL FAIZI
		X	TOPLAM TECIL FAIZI
		E	TOPLAM BORC
		R	TOPLAM BORC
		X	TOPLAM BORC
		E	TECIL FAIZLERI
		R	TECIL FAIZLERI
		X	TECIL FAIZLERI

Table B.10: Measurement Results for Project DVTCL

FUR	FP Name	DM Type	Data Group Desc.
1	Tahakkuk Bilgileri Getir		
		E	TAHAKKUK FISNO

		R	TAHAKKUK BILGISI
		R	BTAHAKKUKTAN KONTROL
		R	BDZTTAHAKKUKTAN KONTROL
		X	VERGI KODU
		R	ONCEKI TECILSATIR BILGISI
		R	ONCEKI TECIL BILGISI
		X	3 AYLIK YAPILABILECEK TECIL MIKTARI
		X	12 AYLIK YAPILABILECEK TECIL MIKTARI
		X	17. MADDE ICIN YAPILABILE-CEK TECIL MIKTARI
		R	TECILANABILGISI
		R	TECILSATIRBILGISI
2	TECIL BILGILERI GIR KAYDET		
		E	3 AYLIK YAPILABILECEK TECIL MIKTARI
		E	12 AYLIK YAPILABILECEK TECIL MIKTARI
		E	17. MADDE ICIN YAPILABILE-CEK TECIL MIKTARI
		E	17. MADDE ICIN TECIL SURESI
		E	3 AYLIK TECIL SECIMI
		E	12 AYLIK TECIL SECIMI
		E	17. MADDE TECIL SECIMI
		X	MESSAGE [01]
		R	KABUL TARIHI DUZELTME KON-TROLU
		X	MESSAGE [02]
		X	MESSAGE [03]
		E	TECIL ANA BILGILERI
		W	TECIL ANA BILGILERI
		E	SATIR BILGILERI
		W	SATIR BILGILERI
		W	TAHAKKUKA TECIL BILGISI
		E	SERVIS BILGISI
		R	SERVIS BILGISI
		E	TECIL DOSYANO
		R	TECIL DOSYANO
		X	TECIL DOSYANO
3	YAZDIR		

		E	PRINTER
		E	DOCUEMENT
		E	TERMINAL BILGISI
		R	TERMINAL BILGISI
		X	TERMINAL BILGISI
		E	VERGI DAIRESI BILGISI
		R	VERGI DAIRESI BILGISI
		X	VERGI DAIRESI BILGISI
		X	TECIL ANA BILGILERI
		X	SATIR BILGILERI
		R	SATICI BILGISI
		X	MUKELLEF BILGILERI
		X	ADRES BILGISI
		R	SATICI BILGISI
		X	SATICI BILGISI

Table B.11: Measurement Results for Project TM

FUR	FP#	FP Name	DM Type	Data Group Desc.
1	1	Arac Bilgisi Al	E	Arac Bilgisi
			R	Arac Detay Bilgisi
			X	Arac Detay Bilgisi
2	2	Borc Hesapla	E	Arac Detay Bilgisi
			R	Tahakkuk Bilgisi
			R	Thksatir Bilgisi
			R	Thksatirhesaplanan Bilgisi
			R	Thktaksit Bilgisi
			R	odemeplaniana Bilgisi
			R	odemeplani Bilgisi
			R	odemeplanisatir Bilgisi
			R	opshesaplanan Bilgisi
			R	Odeme Bilgisi
			R	Tahsilat Bilgisi
			R	Odemesatirhesaplanan Bilgisi
			R	hesaplananvergiodeme Bilgisi
			R	odemesatirhesaplanan Bilgisi
			R	tahsilatekbilgi Bilgisi
			R	odemeekbilgi Bilgisi
			R	tahsilatdetaylari Bilgisi
			R	tahsilataciklama Bilgisi
			R	odemedetay Bilgisi
			R	odemebankabilgileri Bilgisi

			R	odemeemanetbilgileri Bilgisi
			R	odemeindirimbilgileri Bilgisi
			R	odemesaymanlikbilgileri Bilgisi
			R	cekbilgileri Bilgisi
			R	duzeltme Bilgisi
			R	duzeltmebelgeleri Bilgisi
			R	duzeltmeiade Bilgisi
			R	duzeltmeiadesatir Bilgisi
			R	duzeltmesonuclari Bilgisi
			R	tecil414 Bilgisi
			R	tecil414detay Bilgisi
			R	tecil5335 Bilgisi
			R	tecil5335detay Bilgisi
			R	HACIZVARAKA Bilgisi
			R	HACIZVARAKAEKBLG Bilgisi
			R	HACIZVARAKAZIMMET Bilgisi
			R	HACIZVARAKAISLEMLER Bilgisi
			R	HACIZSATIR Bilgisi
			R	HACIZMALLAR Bilgisi
			R	HACIZBILDIRISI Bilgisi
			R	IHTTEMHACIZ Bilgisi
			R	IHTTEMHACIZSATIR Bilgisi
			R	HACIZBILMENKULMAL Bilgisi
			R	HACIZZIMMETISLEMLERI Bilgisi
			R	HACIZBILGAYRIMENKULMAL Bilgisi
			R	HATALIHACIZZIMMETDUSME Bilgisi
			R	HACIZBILDIRISIVARAKALAR Bilgisi
			R	TKPRELATIONS Bilgisi
			R	SRKZTPC Bilgisi
			R	SRKZTPCEKBLG Bilgisi
			R	SRKZOKDTPCORTAK Bilgisi
			R	SRKZOLAYKAYIT Bilgisi
			R	SRKZOKDEKBLG Bilgisi
			R	SRKZOKDSATIR Bilgisi
			R	SRKZEKBLG Bilgisi
			R	SRKZTAPUEKBILGI Bilgisi
			R	SRKZKABAHATTAKSIT Bilgisi
			R	TAKIP Bilgisi
			R	TKPISLEMLER Bilgisi
			R	TKPSATIRLAR Bilgisi
			R	TKPRELATIONS Bilgisi
			R	TKPILANLISTESI Bilgisi

			R	TKPIDARIBILGILERI Bilgisi
			R	HAPSENTAZYIK Bilgisi
			R	HAPSENTAZYIKISLEM Bilgisi
			R	HAPSENTAZYIKTAKIP Bilgisi
			R	dztTahakkuk Bilgisi
			R	dztThksatir Bilgisi
			R	dztThksatirhesaplanan Bilgisi
			R	dztThktaksit Bilgisi
			R	dztodemeplaniana Bilgisi
			R	dztodemeplani Bilgisi
			R	dztodemeplanisatir Bilgisi
			R	dztopshesaplanan Bilgisi
			R	dztOdeme Bilgisi
			R	dztTahsilat Bilgisi
			R	dztOdemesatirhesaplanan Bilgisi
			R	dzthesaplananvergiodeme Bilgisi
			R	dztodemesatirhesaplanan Bilgisi
			R	dzttahsilatekbilgi Bilgisi
			R	dztodemeekbilgi Bilgisi
			R	dzttahsilatdetaylari Bilgisi
			R	dzttahsilataciklama Bilgisi
			R	dztodemedetay Bilgisi
			R	dztodemebankabilgileri Bilgisi
			R	dztodemeemanetbilgileri Bilgisi
			R	dztodemeindirimbilgileri Bilgisi
			R	dztodemesaymanlikbilgileri Bilgisi
			R	dztcekbilgileri Bilgisi
			X	Borc Bilgisi
			X	Tahsilat Bilgisi
3	3	Odeme Bilgisi Al	E	Odeme Bilgisi
			W	Odeme Bilgisi
			X	Onay/Hata
3	4	Kart Bilgisi Al	E	Kart Bilgisi
			W	Kart Bilgisi
			X	Onay/Hata
3	5	Tahsilat Yaz	E	Kart Bilgisi
			E	Tahsilat Bilgisi
			E	Borc Bilgisi
			W	Kart Bilgisi
			W	Tahsilat Bilgisi
			X	Onay/Hata

Table B.12: Measurement Results for Project TN

FUR	FP Name	DM Type	Data Group Desc.
1	signIn	E	User Info
		R	Authorization Info
		X	Authorization Info
2	createVTMRequest	R	Department Information(just once for department dropdown lists)
		X	Department Information(just once for department dropdown lists)
		E	Department Information
		E	Letter of Guarantee Information
		W	Letter of Guarantee Information
		X	Error/Confirmation
3	listVTMRequest	R	Department Information(just once for department dropdown lists)
		X	Department Information(just once for department dropdown lists)
		E	Department Information
		E	Letter of Guarantee Information
		R	Letter of Guarantee Information
		X	Letter of Guarantee Information
4	retrieveVTMRequest	R	Department Information(just once for department dropdown lists)
		X	Department Information(just once for department dropdown lists)
		E	Letter of Guarantee Information
		R	Letter of Guarantee Detail Information
		X	Letter of Guarantee Detail Information
5	UpdateVTMRequest	R	Department Information(just once for department dropdown lists)
		X	Department Information(just once for department dropdown lists)
		E	Department Information
		E	Letter of Guarantee Information
		W	Letter of Guarantee Information
		X	Error/Confirmation
6	deleteVTMRequest	E	Selection of Letter of Guarantee
		W	Letter of Guarantee Information
		X	Error/Confirmation
7	createTMRequestForRealization	R	General Accounting Information(just once for Bank Name dropdown lists)
		X	General Accounting Information(just once for Bank Name dropdown lists)

		R	General Accounting Information(just once for Bank Branch dropdown lists)
		X	General Accounting Information(just once for Bank Branch dropdown lists)
		R	General Accounting Information(just once for Account dropdown lists)
		X	General Accounting Information(just once for Account dropdown lists)
		R	Letter of Guarantee Information(just once for Project dropdown lists)
		X	Letter of Guarantee Information(just once for Project dropdown lists)
		E	Department Information
		E	Letter of Guarantee Information
		E	General Accounting Information
		W	Letter of Guarantee Information
		X	Error/Confirmation
8	listRealizedTM	R	General Accounting Information(just once for Bank Name dropdown lists)
		X	General Accounting Information(just once for Bank Name dropdown lists)
		R	General Accounting Information(just once for Bank Branch dropdown lists)
		X	General Accounting Information(just once for Bank Branch dropdown lists)
		R	General Accounting Information(just once for Account dropdown lists)
		X	General Accounting Information(just once for Account dropdown lists)
		R	Letter of Guarantee Information(just once for Project dropdown lists)
		X	Letter of Guarantee Information(just once for Project dropdown lists)
		E	Department Information
		E	Letter of Guarantee Information
		E	General Accounting Information
		R	Letter of Guarantee Information
		X	Letter of Guarantee Information
9	listWaitingForRealizationTM	R	Department Information(just once for department dropdown lists)
		X	Department Information(just once for department dropdown lists)
		E	Department Information
		E	Letter of Guarantee Information
		R	Letter of Guarantee Information

		X	Letter of Guarantee Information
10	retrieveRealizedTM	R	General Accounting Information(just once for Bank Name dropdown lists)
		X	General Accounting Information(just once for Bank Name dropdown lists)
		R	General Accounting Information(just once for Bank Branch dropdown lists)
		X	General Accounting Information(just once for Bank Branch dropdown lists)
		R	General Accounting Information(just once for Account dropdown lists)
		X	General Accounting Information(just once for Account dropdown lists)
		R	Letter of Guarantee Information(just once for Project dropdown lists)
		X	Letter of Guarantee Information(just once for Project dropdown lists)
		E	Selection of Letter of Guarantee Information
		R	Letter of Guarantee Detail Information
		X	Letter of Guarantee Detail Information
11	realizeTM	E	Letter of Guarantee Information
		E	General Accounting Information
		W	Letter of Guarantee Information
		W	General Accounting Information
		R	Letter of Guarantee Information
		R	General Accounting Information
		X	Letter of Guarantee Information
		X	General Accounting Information
12	printTM	E	Letter of Guarantee Information
		E	General Accounting Information
		R	Letter of Guarantee Information
		R	General Accounting Information
		X	Letter of Guarantee Information
		X	General Accounting Information
13	createAccountingDataForRealizedTM	E	Letter of Guarantee Information
		E	General Accounting Information
		W	Daily Voucher Information
		X	Error/Confirmation
14	deliverTM	E	Selection of Letter of Guarantee Information
		W	Letter of Guarantee Information
		R	Letter of Guarantee Information

		X	Letter of Guarantee Information
15	listCommisionRates	R	General Accounting Information(just once for Bank Name dropdown lists)
		X	General Accounting Information(just once for Bank Name dropdown lists)
		E	Letter of Guarantee Information
		R	Letter of Guarantee Information
		X	Letter of Guarantee Information
16	createAccountingDataForCommissionRate	E	Letter of Guarantee Information
		E	General Accounting Information
		W	Daily Voucher Information
		X	Error/Confirmation
17	listExpiredTM	E	Letter of Guarantee Information
		R	Letter of Guarantee Information
		X	Letter of Guarantee Information
18	returnTM	E	Letter of Guarantee Information
		E	General Accounting Information
		W	Daily Voucher Information
		X	Error/Confirmation
19	sendEmail	E	Letter of Guarantee Information
		E	Department Information
		E	User Information
		X	E-Mail
20	lengthenExpireDateofTM	E	Selection of Letter of Guarantee Information
		W	Letter of Guarantee Information
		X	Error/Confirmation
21	createATM	R	General Accounting Information(just once for Bank Name dropdown lists)
		X	General Accounting Information(just once for Bank Name dropdown lists)
		R	Department Information(just once for department dropdown lists)
		X	Department Information(just once for department dropdown lists)
		E	Department Information
		E	Letter of Guarantee Information
		E	General Accounting Information
		W	Letter of Guarantee Information
		X	Error/Confirmation
22	listATM	R	General Accounting Information(just once for Bank Name dropdown lists)
		X	General Accounting Information(just once for Bank Name dropdown lists)

		R	Department Information(just once for department dropdown lists)
		X	Department Information(just once for department dropdown lists)
		E	Department Information
		E	Letter of Guarantee Information
		E	General Accounting Information
		R	Letter of Guarantee Information
		X	Letter of Guarantee Information
23	retriveATM	R	General Accounting Information(just once for Bank Name dropdown lists)
		X	General Accounting Information(just once for Bank Name dropdown lists)
		R	Department Information(just once for department dropdown lists)
		X	Department Information(just once for department dropdown lists)
		E	Selection of Letter of Guarantee Information
		R	Letter of Guarantee Detail Information
		X	Letter of Guarantee Detail Information
24	UpdateATM	R	General Accounting Information(just once for Bank Name dropdown lists)
		X	General Accounting Information(just once for Bank Name dropdown lists)
		R	Department Information(just once for department dropdown lists)
		X	Department Information(just once for department dropdown lists)
		E	Department Information
		E	Letter of Guarantee Information
		E	General Accounting Information
		W	Letter of Guarantee Information
		X	Error/Confirmation
25	deleteATM	E	Selection of Letter of Guarantee
		W	Letter of Guarantee Information
		X	Error/Confirmation
26	createAccountingDataForATM	E	Letter of Guarantee Information
		E	General Accounting Information
		W	Daily Voucher Information
		X	Error/Confirmation
27	deliverATM	E	Selection of Letter of Guarantee Information

		W	Letter of Guarantee Information
		R	Letter of Guarantee Information
		X	Letter of Guarantee Information

APPENDIX C

Appendix C: Movie Manager Application Specifications and COSMIC Model

A SAMPLE CASE FOR COSMIC v3.0.1 - MOVIE MANAGER

Calculate the functional size of the requirements of the Movie Manager application with the COSMIC FSM. Indicate any relevant assumptions you made.

Movies: The application shall keep record of movies. Movie title, year of production, production company & genre of the movies should be kept for movies. The genre can be of the following type: Comedy, thriller, animation, documentary, science fiction, action, horror, drama, musical and western.

Movies shall also have director, producer, writer and cast information where all can have more than one record each.

Person: The application shall keep record of persons related to movies. Person information shall include, name of the person, date of birth and place of birth.

A person might be acting as an actress/actor, or might be a producer, writer or director of a movie. It is also possible for a person to be all or a combination of these (both writer and director, etc.) in a movie.

§ If a person is an actress/actor in a movie, the application shall also maintain the character name in the movie.

§ If a person is the producer of a movie. It shall also be noted whether he/she is the coproducer, executive producer or just the producer.

§ If a person is the writer of a movie. It shall also be noted whether he/she is the story writer, screenplay writer or both.

§ If a person is the director of a movie no additional attributes need to be maintained.

The functional requirements to be measured:

1. The application shall enable the entry and update of persons.

For updates, the application shall first provide a list of all persons. Once a person is selected, the application shall display the details of the person on an editable form on which the updated information will be entered.

2. The application shall enable the entry of movie information. The movie information shall be entered in two steps via a wizard:

   In the first step; the movie title, year of production, production company & genre of the movie will be entered. Genre options shall be displayed in a list box and will be selected from this list.

   In the second step; the director, producer, writer and cast information will be entered. Persons shall be selected from relevant list boxes which list the persons recorded in the system.

   The user shall be allowed to quit the wizard without saving it.

3. The application shall enable an enquiry of movies over the title and the year the movie is produced. The application shall list the title and the year of the movies that match the query parameters.

   Once user selects a specific movie, details of the movie shall be listed. The output shall include the following information:

a. title, year of production, production company, genre

b. director(s) (name),

c. producer(s) [co/executive] (name),

d. writer(s) [story/screenplay/story & screenplay] (name),

e. cast (actor name, character name)

4. The application shall enable an enquiry of persons by their name. The application shall list the name of the persons that match with the query parameter. Once the user selects a specific person, details shall be listed. The output shall include the following information:

a. name, date of birth & place of birth

b. movies directed (Title and production year of the movie),

c. movies produced (Title and production year of the movie ,with co/executive/producer indicated),

d. movies written (Title and production year of the movie , with story/screenplay/story & screenplay indicated),

e. movies acted (Title and production year of the movie , with character name indicated)

5. The application shall enable the deletion of movies. First the application shall provide a list of all movies.

Once a movie is selected, the application shall delete all related information and display the updated movie list as a confirmation.

6. The application shall enable the deletion of persons. First the application shall provide a list of all persons.

Once a person is selected, the application shall delete all related person information. The relevant person

information should be removed from associated movie roles (writer, actor, etc.) with one exception that if the user is a director of a movie, it shall not be deleted and a warning message should be displayed stating that the person to be deleted is a director and will not be deleted. Otherwise the user should return to the updated persons list as a confirmation. Multiple users can be selected for deletion.

7. If not specified explicitly; for write, delete and update operations the application will produce error/confirmation messages.

Entity Relationship Diagram:

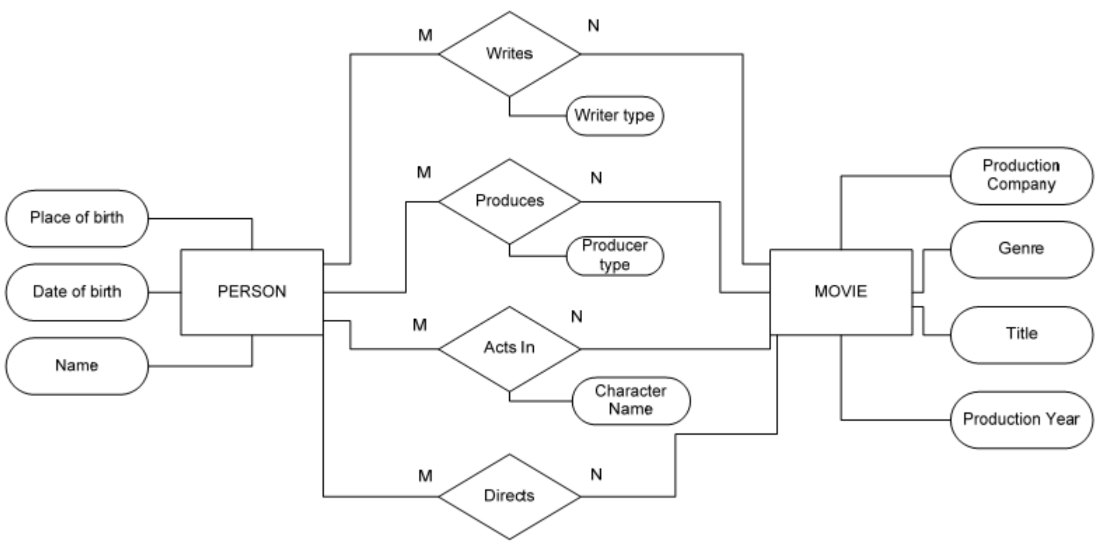


Table C.1: Movie Manager COSMIC Model

FUR	FP#	FP Name	DM Type	Data Group Desc.
1	1	Add Person	E	Person info
			W	Person info
			X	Error/Confirmation
1	2	List Persons	E	List persons request
			R	Persons info
			X	Persons info
1	3	Retrieve Person	E	Retrive person details request
			R	Person details info
			X	Person details info
1	4	Update Person	E	Person info
			W	Person info
			X	Error/Confirmation
2	5	Add Movie	E	Movie Info (title, year, prod.company & genre)
			R	Person Info (just once for writer, producer, cast and director dropdown lists)
			X	Person Info (just once for writer, producer, cast and director dropdown lists)
			E	Writer info
			E	Producer info
			E	Cast info
			E	Director info
			W	Movie Info
			W	Writer info
			W	Producer info
			W	Cast info
			W	Director info
			X	Error/Confirmation
3	6	Query Movie	E	Query Parameters (movie title, year)
			R	Movie Info
			X	Movie Info (title, year)(OOI:set of movies that match the criteria)
3	7	List Movie Details	E	Selection of the movie
			R	Movie Info
			R	Writer info
			R	Producer info
			R	Cast info

			R	Director info
			R	Person info (Name) (counted just once for writer, producer, cast and director)
			X	Movie Info
			X	Writer info
			X	Producer info
			X	Cast info
			X	Director info
			X	Person info (Name) (counted just once for writer, producer, cast and director)
4	8	Query Person	E	Query Parameters (person name)
			R	Person Info
			X	Person Info (name)(OOI:Set of persons .t.m.t.c)
4	9	List Person Details	E	Selection of the person
			R	Person Info
			R	Writer info
			R	Producer info
			R	Cast info
			R	Director info
			R	Movie info (counted just once for writer, producer, cast and director)
			X	Person Info
			X	Writer info
			X	Producer info
			X	Cast info
			X	Director info
			X	Movie info (Title, year) (counted just once for writer, producer, cast and director)
5	10	List Movies	E	Request for a list of movies
			R	Movie info
			X	Movie info (title, year)
5	11	Delete Movie	E	Selection of the movie
			W	Movie Info
			R	Writer info (OOI:Set of Writers that are associated with a selected movie)
			R	Producer info(similar to writerinfo)
			R	Cast info(similar to writerinfo)
			R	Director info(similar to writerinfo)
			W	Writer info
			W	Producer info
			W	Cast info

			W	Director info
			R	Movie Info
			X	Movie info (title, year)
6	12	Delete Person	E	Selection of the person
			R	Writer info (OOI:Set of movie Writers that are associated with a selected person)
			R	Producer info(similar to writerinfo)
			R	Cast info(similar to writerinfo)
			R	Director info(similar to writerinfo)
			W	Writer info
			W	Producer info
			W	Cast info
			X	Confirm/Error
			X	Person

APPENDIX D

Appendix D: Case Study 2- Measurement Results

Table D.1: MM COSMIC FSM Results for Measurer # 1

FUR	FP#	FP Name	Fp?	DM Type	Data Group Desc.	Comments
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	
						1X Confirmation & Error Seperated
1	2	List Persons	0	E	List persons request	
				R	Persons info	Add - List - Retrieve - Update taken as one FP
				X	Persons info	NO X for listing
1	3	Retrieve Person	0	E	Retrive person details request	Add - List - Retrieve - Update taken as one FP
				R	Person details info	Add - List - Retrieve - Update taken as one FP
				X	Person details info	Add - List - Retrieve - Update taken as one FP
1	4	Update Person	0	E	Person info	Add - List - Retrieve - Update taken as one FP
				W	Person info	Add - List - Retrieve - Update taken as one FP
				X	Error/Confirmation	Add - List - Retrieve - Update taken as one FP
2	5	Add Movie	0	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	

				X	Person Info (just once for writer, producer, cast and director dropdown lists)	No X for listing
				E	Writer info	Entry for different attributes are combined (counted as 1)
				E	Producer info	Entry for different attributes are combined (counted as 1)
				E	Cast info	Entry for different attributes are combined (counted as 1)
				E	Director info	Entry for different attributes are combined (counted as 1)
				W	Movie Info	
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				X	Error/Confirmation	
						1W for non-existent OOI (Genre) 1E for non-existent OOI (Genre) 1X Confirmation & Error Seperated 1W for attribute abstract grouping
3	6	Query Movie	0	E	Query Parameters (movie title, year)	
				R	Movie Info	Selective operations handled seperately
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	No X for Query
						1R Selective operations handled seperately
3	7	List Movie Details	1	E	Selection of the movie	Query and Detail Retreival are taken as one FP
				R	Movie Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	
				X	Movie Info	No X for Query
				X	Writer info	No X for Query
				X	Producer info	No X for Query
				X	Cast info	No X for Query

				X	Director info	No X for Query
				X	Person info (Name) (counted just once for writer, producer, cast and director)	No X for Query
						3R Person Info Counted for every type 1X Error Message
4	8	Query Person	0	E	Query Parameters (person name)	
				R	Person Info	
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	
4	9	List Person Details	0	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	
5	10	List Movies	1	E	Request for a list of movies	
				R	Movie info	
				X	Movie info (title, year)	
						1X Confirmation Message
5	11	Delete Movie	1	E	Selection of the movie	No E for Selection
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete

				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				R	Movie Info	
				X	Movie info (title, year)	No X for Confirmation / Error after Delete
						1W Problem Specific (No need to delete Genre upon deleting a movie)
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	R for different sub-types are combined (counted as 1)
				R	Producer info(similar to writerinfo)	R for different sub-types are combined (counted as 1)
				R	Cast info(similar to writerinfo)	R for different sub-types are combined (counted as 1)
				R	Director info(similar to writerinfo)	R for different sub-types are combined (counted as 1)
				W	Writer info	
				W	Producer info	
				W	Cast info	
				X	Confirm/Error	
				X	Person	No X after Delete - Expected: Return to list as confirmation
						1R Assumption of listing before Delete 1W Trying W before illegal operation (Problem Specific) - Deleting Director

Table D.2: MM COSMIC FSM Results for Measurer # 2

FUR	**FP#**	**FP Name**	**Fp?**	**DM Type**	**Data Group Desc.**	**Comments**
1	1	Add Person	1	E	Person info	
				W	Person info	

				X	Error/Confirmation	No X for Confirmation / Error after Add
1	2	List Persons	1	E	List persons request	
				R	Persons info	
				X	Persons info	
1	3	Retrieve Person	0	E	Retrive person details request	List & Retrieve Combined
				R	Person details info	List & Retrieve Combined
				X	Person details info	List & Retrieve Combined
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Confirmation/Error after Update
						1R Assumed R before Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices
				E	Writer info	
				E	Producer info	
				E	Cast info	
				E	Director info	Trivial Mistake
				W	Movie Info	
				W	Writer info	Write for different attributes are combined (counted as 1)
				W	Producer info	Write for different attributes are combined (counted as 1)
				W	Cast info	Write for different attributes are combined (counted as 1)
				W	Director info	Write for different attributes are combined (counted as 1)
				X	Error/Confirmation	
						1E For Separate attribute. (Genre) ? 1R For Genre
3	6	Query Movie	1	E	Query Parameters (movie title, year)	
				R	Movie Info	

				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	
3	7	List Movie Details	1	E	Selection of the movie	
				R	Movie Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	No R due to Missing OOI
				X	Movie Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Person info (Name) (counted just once for writer, producer, cast and director)	No X due to Missing OOI
4	8	Query Person	1	E	Query Parameters (person name)	
				R	Person Info	
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	No R due to Missing OOI
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	

				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to Missing OOI
						1E after list
5	10	List Movies	1	E	Request for a list of movies	
				R	Movie info	
				X	Movie info (title, year)	
						4E Inclusion of extra OOI? 4X Inclusion of extra OOI?
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1W Problem Specific (No need to delete Genre upon deleting a movie) 1X for Error / Confirmation
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	No R before Delete
				R	Producer info(similar to writerinfo)	No R before Delete
				R	Cast info(similar to writerinfo)	No R before Delete
				R	Director info(similar to writerinfo)	No R before Delete
				W	Writer info	

				W	Producer info	
				W	Cast info	
				X	Confirm/Error	
				X	Person	
						”List Persons” Repeated as FP before ”Delete Person” 1W Separate W for abstract group (person)

Table D.3: MM COSMIC FSM Results for Measurer # 3

FUR	FP#	FP Name	Fp?	DM Type	Data Group Desc.	Comments
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Add
1	2	List Persons	1	E	List persons request	
				R	Persons info	
				X	Persons info	
1	3	Retrieve Person	1	E	Retrive person details request	
				R	Person details info	
				X	Person details info	No X After Retrieve
						1W Write after Retrieve
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	Add Movie Separated to two FPs.
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices
				E	Writer info	
				E	Producer info	
				E	Cast info	
				E	Director info	

				W	Movie Info	
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				X	Error/Confirmation	
						1E For Separate attribute. (Genre) ? 1W For Separate attribute. (Genre) ? 1E For Abstract Group Selection (Person) 1W ForAbstract Group Write (Person)
3	6	Query Movie	1	E	Query Parameters (movie title, year)	
				R	Movie Info	
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	
3	7	List Movie Details	1	E	Selection of the movie	
				R	Movie Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	
				X	Movie Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Person info (Name) (counted just once for writer, producer, cast and director)	
						1R For Genre 1X For Genre
4	8	Query Person	1	E	Query Parameters (person name)	
				R	Person Info	
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	

4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	
5	10	List Movies	1	E	Request for a list of movies	
				R	Movie info	
				X	Movie info (title, year)	
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error / Confirmation
6	12	Delete Person		E	Selection of the person	

				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	No R before Delete
				R	Producer info(similar to writerinfo)	No R before Delete
				R	Cast info(similar to writerinfo)	No R before Delete
				R	Director info(similar to writerinfo)	No R before Delete
				W	Writer info	
				W	Producer info	
				W	Cast info	
				X	Confirm/Error	
				X	Person	
						1R 1X 1E"List Persons" Repeated as FP before "Delete Person" 1W For Abstract Group (Person)

Table D.4: MM COSMIC FSM Results for Measurer # 4

FUR	**FP#**	**FP Name**	**Fp?**	**DM Type**	**Data Group Desc.**	**Comments**
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
1	2	List Persons	0	E	List persons request	Missing Functionality
				R	Persons info	Missing Functionality
				X	Persons info	Missing Functionality
1	3	Retrieve Person	0	E	Retrive person details request	Missing Functionality
				R	Person details info	Missing Functionality
				X	Person details info	Missing Functionality
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Update

						1R Assumed R Before Update 1E Assumed R Before Update 1X Assumed R Before Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices
				E	Writer info	Trivial Mistake
				E	Producer info	
				E	Cast info	
				E	Director info	
				W	Movie Info	
				W	Writer info	Trivial Mistake
				W	Producer info	
				W	Cast info	
				W	Director info	
				X	Error/Confirmation	No X for Error / Confirmation after Add
						1E For Separate attribute. (Genre) ?
3	6	Query Movie	0	E	Query Parameters (movie title, year)	Query and Details Listing Combined
				R	Movie Info	
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	
3	7	List Movie Details	1	E	Selection of the movie	
				R	Movie Info	
				R	Writer info	Trivial Mistake
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Movie Info	
				X	Writer info	Trivial Mistake
				X	Producer info	
				X	Cast info	
				X	Director info	

				X	Person info (Name) (counted just once for writer, producer, cast and director)	No X due to missing OOI
						1R For Genre 1X For Genre 1X For Error
4	8	Query Person	0	E	Query Parameters (person name)	Query and Details Listing Combined
				R	Person Info	
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to missing OOI
						1X for Error/Confirmation
5	10	List Movies	0	E	Request for a list of movies	
				R	Movie info	
				X	Movie info (title, year)	
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete

				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	No W to delete related date on Delete
				W	Producer info	No W to delete related date on Delete
				W	Cast info	No W to delete related date on Delete
				W	Director info	No W to delete related date on Delete
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	No R before Delete
				R	Producer info(similar to writerinfo)	No R before Delete
				R	Cast info(similar to writerinfo)	No R before Delete
				R	Director info(similar to writerinfo)	No R before Delete
				W	Writer info	
				W	Producer info	
				W	Cast info	
				X	Confirm/Error	
				X	Person	No X after Delete - Expected: Return to list as confirmation
						1W Trying W before illegal operation (Problem Specific) - Deleting Director

Table D.5: MM COSMIC FSM Results for Measurer # 5

FUR	**FP#**	**FP Name**	**Fp?**	**DM Type**	**Data Group Desc.**	**Comments**
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
1	2	List Persons	0	E	List persons request	
				R	Persons info	

				X	Persons info	
1	3	Retrieve Person	0	E	Retrive person details request	Update & Retrieve Combined
				R	Person details info	
				X	Person details info	
1	4	Update Person	1	E	Person info	No E for Update
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices
				E	Writer info	Combined as ”Selected Role Info”
				E	Producer info	Combined as ”Selected Role Info”
				E	Cast info	Combined as ”Selected Role Info”
				E	Director info	Combined as ”Selected Role Info”
				W	Movie Info	
				W	Writer info	Combined as ”Selected Role Info”
				W	Producer info	Combined as ”Selected Role Info”
				W	Cast info	Combined as ”Selected Role Info”
				W	Director info	Combined as ”Selected Role Info”
				X	Error/Confirmation	
						1W for Genre 1W for ”Lookup Info”?
3	6	Query Movie	0	E	Query Parameters (movie title, year)	Missing Functionality
				R	Movie Info	Missing Functionality
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	Missing Functionality
3	7	List Movie Details	1	E	Selection of the movie	
				R	Movie Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	

				R	Person info (Name) (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Movie Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Person info (Name) (counted just once for writer, producer, cast and director)	No X due to missing OOI
						1R for Genre 1W for Genre
4	8	Query Person	0	E	Query Parameters (person name)	Query and Listing Combined
				R	Person Info	X Without R in Query
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to missing OOI
5	10	List Movies	0	E	Request for a list of movies	Missing Functionality
				R	Movie info	
				X	Movie info (title, year)	

5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error / Confirmation
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	No R before Delete
				R	Producer info(similar to writerinfo)	No R before Delete
				R	Cast info(similar to writerinfo)	No R before Delete
				R	Director info(similar to writerinfo)	No R before Delete
				W	Writer info	
				W	Producer info	
				W	Cast info	
				X	Confirm/Error	
				X	Person	No X after Delete - Expected: Return to list as confirmation

Table D.6: MM COSMIC FSM Results for Measurer # 6

FUR	**FP#**	**FP Name**	**Fp?**	**DM Type**	**Data Group Desc.**	**Comments**
1	1	Add Person	1	E	Person info	

				W	Person info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
1	2	List Persons	1	E	List persons request	
				R	Persons info	
				X	Persons info	
1	3	Retrieve Person	1	E	Retrive person details request	
				R	Person details info	
				X	Person details info	
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices - Assumed Re-use of ”list People”
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices - Assumed Re-use of ”list People”
				E	Writer info	
				E	Producer info	
				E	Cast info	
				E	Director info	
				W	Movie Info	
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				X	Error/Confirmation	
						1R for Genre 1X for Genre
3	6	Query Movie	1	E	Query Parameters (movie title, year)	
				R	Movie Info	
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	
3	7	List Movie Details	0	E	Selection of the movie	Query and Details Listing Combined

				R	Movie Info	Query and Details Listing Combined
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	Assumed Re-use of ”list People”
				X	Movie Info	Query and Details Listing Combined
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Person info (Name) (counted just once for writer, producer, cast and director)	Assumed Re-use of ”list People”
4	8	Query Person	0	E	Query Parameters (person name)	Missing Functionality
				R	Person Info	Missing Functionality
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	Missing Functionality
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to missing OOI

5	10	List Movies	1	E	Request for a list of movies	
				R	Movie info	
				X	Movie info (title, year)	
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	
				R	Producer info(similar to writerinfo)	
				R	Cast info(similar to writerinfo)	
				R	Director info(similar to writerinfo)	
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error / Confirmation
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	Assumed Re-use of ”list People”
				R	Producer info(similar to writerinfo)	Assumed Re-use of ”list People”
				R	Cast info(similar to writerinfo)	Assumed Re-use of ”list People”
				R	Director info(similar to writerinfo)	Assumed Re-use of ”list People”
				W	Writer info	
				W	Producer info	
				W	Cast info	
				X	Confirm/Error	
				X	Person	No X after Delete - Expected: Return to list as confirmation
						1W Person Info

Table D.7: MM COSMIC FSM Results for Measurer # 7

FUR	FP#	FP Name	Fp?	DM Type	Data Group Desc.	Comments
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
1	2	List Persons	1	E	List persons request	
				R	Persons info	
				X	Persons info	
1	3	Retrieve Person	1	E	Retrive person details re-quest	
				R	Person details info	
				X	Person details info	
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	Missing Functionality
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	Missing Functionality
				E	Writer info	Missing Functionality
				E	Producer info	Missing Functionality
				E	Cast info	Missing Functionality
				E	Director info	Missing Functionality
				W	Movie Info	
				W	Writer info	Missing Functionality
				W	Producer info	Missing Functionality
				W	Cast info	Missing Functionality
				W	Director info	Missing Functionality
				X	Error/Confirmation	Missing Functionality
						1R for Genre 1X for Genre
3	6	Query Movie	1	E	Query Parameters (movie title, year)	
				R	Movie Info	

				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	
3	7	List Movie Details	1	E	Selection of the movie	
				R	Movie Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Movie Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Person info (Name) (counted just once for writer, producer, cast and director)	No X due to missing OOI
4	8	Query Person	1	E	Query Parameters (person name)	Missing Functionality
				R	Person Info	Missing Functionality
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	Missing Functionality
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	

				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to missing OOI
5	10	List Movies	1	E	Request for a list of movies	
				R	Movie info	
				X	Movie info (title, year)	
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	No W to update related date at Delete
				W	Producer info	No W to update related date at Delete
				W	Cast info	No W to update related date at Delete
				W	Director info	No W to update related date at Delete
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error / Confirmation
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	Combined in ”Person Info”
				R	Producer info(similar to writerinfo)	Combined in ”Person Info”
				R	Cast info(similar to writerinfo)	Combined in ”Person Info”
				R	Director info(similar to writerinfo)	Combined in ”Person Info”
				W	Writer info	Combined in ”Person Info”
				W	Producer info	Combined in ”Person Info”
				W	Cast info	Combined in ”Person Info”
				X	Confirm/Error	Combined in ”Person Info”

				X	Person	No X after Delete - Expected: Return to list as confirmation
						1E Re-count for listing people 1X Re-count for listing people

Table D.8: MM COSMIC FSM Results for Measurer # 8

FUR	FP#	FP Name	Fp?	DM Type	Data Group Desc.	Comments
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
1	2	List Persons	1	E	List persons request	
				R	Persons info	
				X	Persons info	Note:"List and Update may be merged"
1	3	Retrieve Person	0	E	Retrive person details request	List and Retrieve Combined
				R	Person details info	List and Retrieve Combined
				X	Person details info	List and Retrieve Combined
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	Missing Functionality
				E	Writer info	No E Before W / Combined as E Movie Info
				E	Producer info	No E Before W / Combined as E Movie Info
				E	Cast info	No E Before W / Combined as E Movie Info

				E	Director info	No E Before W / Combined as E Movie Info
				W	Movie Info	Note: ”optional and may not be used with another approach”
				W	Writer info	Note: ”optional and may not be used with another approach”
				W	Producer info	Note: ”optional and may not be used with another approach”
				W	Cast info	Note: ”optional and may not be used with another approach”
				W	Director info	Note: ”optional and may not be used with another approach”
				X	Error/Confirmation	
						1W for Genre
3	6	Query Movie	1	E	Query Parameters (movie title, year)	Note:”List and Update may be merged”
				R	Movie Info	
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	
3	7	List Movie Details	1	E	Selection of the movie	Used subtypes when reading, and only parent type when X’ing
				R	Movie Info	
				R	Writer info	Note: ”optional and may not be used with another approach”
				R	Producer info	Note: ”optional and may not be used with another approach”
				R	Cast info	Note: ”optional and may not be used with another approach”
				R	Director info	Note: ”optional and may not be used with another approach”
				R	Person info (Name) (counted just once for writer, producer, cast and director)	Note: ”optional and may not be used with another approach”
				X	Movie Info	Note: ”optional and may not be used with another approach”
				X	Writer info	Combined as ”Movie Info”
				X	Producer info	Combined as ”Movie Info”
				X	Cast info	Combined as ”Movie Info”
				X	Director info	Combined as ”Movie Info”

				X	Person info (Name) (counted just once for writer, producer, cast and director)	Combined as ”Movie Info”
						1R For Genre
4	8	Query Person	1	E	Query Parameters (person name)	
				R	Person Info	
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	
4	9	List Person Details	1	E	Selection of the person	Used subtypes when reading, and only parent type when X’ing
				R	Person Info	No R due to not including abstract group (person)
				R	Writer info	Note: ”optional and may not be used with another approach”
				R	Producer info	Note: ”optional and may not be used with another approach”
				R	Cast info	Note: ”optional and may not be used with another approach”
				R	Director info	Note: ”optional and may not be used with another approach”
				R	Movie info (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Person Info	
				X	Writer info	Combined as ”Person Info”
				X	Producer info	Combined as ”Person Info”
				X	Cast info	Combined as ”Person Info”
				X	Director info	Combined as ”Person Info”
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to missing OOI
5	10	List Movies	0	E	Request for a list of movies	List and Retrieve Combined
				R	Movie info	List and Retrieve Combined
				X	Movie info (title, year)	List and Retrieve Combined
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	

				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	Note: ”optional and may not be used with another approach”
				W	Producer info	Note: ”optional and may not be used with another approach”
				W	Cast info	Note: ”optional and may not be used with another approach”
				W	Director info	Note: ”optional and may not be used with another approach”
				R	Movie Info	Note: ”optional and may not be used with another approach”
				X	Movie info (title, year)	
						1W Problem Specific (No need to delete Genre upon deleting a movie)
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	Combined in ”Person Info”
				R	Producer info(similar to writerinfo)	Combined in ”Person Info”
				R	Cast info(similar to writerinfo)	Combined in ”Person Info”
				R	Director info(similar to writerinfo)	Combined in ”Person Info”
				W	Writer info	Note: ”optional and may not be used with another approach”
				W	Producer info	Note: ”optional and may not be used with another approach”
				W	Cast info	Note: ”optional and may not be used with another approach”
				X	Confirm/Error	
				X	Person	No X after Delete - Expected: Return to list as confirmation

Table D.9: MM COSMIC FSM Results for Measurer # 9

FUR	FP#	FP Name	Fp?	DM Type	Data Group Desc.	Comments
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
1	2	List Persons	1	E	List persons request	
				R	Persons info	X Without R
				X	Persons info	
1	3	Retrieve Person	0	E	Retrive person details request	List and retrieve Combined
				R	Person details info	
				X	Person details info	
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices - Assumed Re-use of ”list People”
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	No listing of Choices - Assumed Re-use of ”list People”
				E	Writer info	
				E	Producer info	
				E	Cast info	
				E	Director info	
				W	Movie Info	
				W	Writer info	Combined as W Movie Info
				W	Producer info	Combined as W Movie Info
				W	Cast info	Combined as W Movie Info
				W	Director info	Combined as W Movie Info
				X	Error/Confirmation	No X for Error/Confirmation after Add
						1E For Genre 1E For OOI: ”Producer Type” 1E For OOI: ”Cast Character” 1E For ”Quit”

3	6	Query Movie	0	E	Query Parameters (movie title, year)	Query and Details Listing Combined
				R	Movie Info	Query and Details Listing Combined
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	Query and Details Listing Combined
3	7	List Movie Details	1	E	Selection of the movie	Counted under ”Query” - Query and Details Listing Combined
				R	Movie Info	Counted under ”Query” - Query and Details Listing Combined Combined as R for Data Group ”Movie,Person”
				R	Writer info	Combined as ”Movie Info”
				R	Producer info	Combined as ”Movie Info”
				R	Cast info	Combined as ”Movie Info”
				R	Director info	Combined as ”Movie Info”
				R	Person info (Name) (counted just once for writer, producer, cast and director)	Combined as R for Data Group ”Movie,Person”
				X	Movie Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Person info (Name) (counted just once for writer, producer, cast and director)	No X due to missing OOI
4	8	Query Person	0	E	Query Parameters (person name)	Query and Details Listing Combined
				R	Person Info	Query and Details Listing Combined
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	Query and Details Listing Combined
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	Combined as ”Person Info”
				R	Producer info	Combined as ”Person Info”
				R	Cast info	Combined as ”Person Info”
				R	Director info	Combined as ”Person Info”

				R	Movie info (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to missing OOI
5	10	List Movies	0	E	Request for a list of movies	List and Delete combined
				R	Movie info	List and Delete combined
				X	Movie info (title, year)	List and Delete combined
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are asso-ciated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to wri-terinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	No W to update related date at Delete
				W	Producer info	No W to update related date at Delete
				W	Cast info	No W to update related date at Delete
				W	Director info	No W to update related date at Delete
				R	Movie Info	No R after Delete (to be used in Con-firmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error / Confirmation
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are as-sociated with a selected person)	
				R	Producer info(similar to writerinfo)	

				R	Cast info(similar to writerinfo)	
				R	Director info(similar to writerinfo)	
				W	Writer info	1W for ”Delete Write, Actor Info”
				W	Producer info	
				W	Cast info	1W for ”Delete Write, Actor Info”
				X	Confirm/Error	
				X	Person	No X after Delete - Expected: Return to list as confirmation
						1W Person Info 1X For List all Persons 1R ”If person is director Y/N” 1E For user feedback ”Y/N”

Table D.10: MM COSMIC FSM Results for Measurer # 10

FUR	**FP#**	**FP Name**	**Fp?**	**DM Type**	**Data Group Desc.**	**Comments**
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
1	2	List Persons	1	E	List persons request	
				R	Persons info	
				X	Persons info	
1	3	Retrieve Person	1	E	Retrive person details request	
				R	Person details info	
				X	Person details info	
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	
						1R Person Info
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	

				X	Person Info (just once for writer, producer, cast and director dropdown lists)	
				E	Writer info	Used R Instead
				E	Producer info	Used R Instead
				E	Cast info	Used R Instead
				E	Director info	Used R Instead
				W	Movie Info	
				W	Writer info	Combined as W Movie Info
				W	Producer info	Combined as W Movie Info
				W	Cast info	Combined as W Movie Info
				W	Director info	Combined as W Movie Info
				X	Error/Confirmation	
						1R for Genre 1X for Genre 1R for Movie Info 1R for Director Info 1R for Producer Info 1R for Writer Info 1R for Cast Info
3	6	Query Movie	0	E	Query Parameters (movie title, year)	Query and Details Listing Combined
				R	Movie Info	Query and Details Listing Combined
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	Query and Details Listing Combined
3	7	List Movie Details	1	E	Selection of the movie	
				R	Movie Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	
				X	Movie Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Person info (Name) (counted just once for writer, producer, cast and director)	
						1R for Genre 1W for ”Lookup Info” 1X for ”Lookup Info”

4	8	Query Person	0	E	Query Parameters (person name)	Query and Details Listing Combined
				R	Person Info	Query and Details Listing Combined
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	Query and Details Listing Combined
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	No Separate X for Director
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to missing OOI
						1W for ”Lookup Info” 1X for ”Lookup Info”
5	10	List Movies	1	E	Request for a list of movies	
				R	Movie info	
				X	Movie info (title, year)	
						1E, 1R, 1X for additional FP ”Retrieve Movie For Delete”
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	No W for Movie Info at Movie Delete
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete

				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error / Confirmation 1W Problem Specific (No need to delete Genre upon deleting a movie)
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	
				R	Producer info(similar to writerinfo)	
				R	Cast info(similar to writerinfo)	
				R	Director info(similar to writerinfo)	
				W	Writer info	
				W	Producer info	
				W	Cast info	
				X	Confirm/Error	
				X	Person	No X after Delete - Expected: Return to list as confirmation
						1X Warning and Confirmation are separatedand counted as 2X 1E, 1R, 1X for additional ”List Persons” 1E, 1R, 1X for additional ”Retrieve the person for update”

Table D.11: MM COSMIC FSM Results for Measurer # 11

FUR	**FP#**	**FP Name**	**Fp?**	**DM Type**	**Data Group Desc.**	**Comments**
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
1	2	List Persons	1	E	List persons request	

				R	Persons info	
				X	Persons info	
1	3	Retrieve Person	1	E	Retrive person details request	
				R	Person details info	
				X	Person details info	
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	Abstract Group OOI Problem
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	Abstract Group OOI Problem
				E	Writer info	
				E	Producer info	
				E	Cast info	
				E	Director info	
				W	Movie Info	
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
						1E for Genre 1R for Genre 1X for Genre 1W for Genre
3	6	Query Movie	0	E	Query Parameters (movie title, year)	Query and Details Listing Combined
				R	Movie Info	Query and Details Listing Combined
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	Query and Details Listing Combined
3	7	List Movie Details	1	E	Selection of the movie	Query and Details Listing Combined
				R	Movie Info	Query and Details Listing Combined
				R	Writer info	
				R	Producer info	

				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Movie Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Person info (Name) (counted just once for writer, producer, cast and director)	No X due to missing OOI
						1R For Genre
4	8	Query Person	0	E	Query Parameters (person name)	Query and Details Listing Combined
				R	Person Info	Query and Details Listing Combined
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	Query and Details Listing Combined
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	No R due to missing OOI
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to missing OOI
5	10	List Movies	1	E	Request for a list of movies	
				R	Movie info	

				X	Movie info (title, year)	
						1E,5R, 5X For additional FP ”Retrieve Selected Movie”
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error / Confirmation
6	12	Delete Person	1	E	Selection of the person	FP Identified (Zaman Yetmemis gibi)
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	FP Identified (Zaman Yetmemis gibi)
				R	Producer info(similar to writerinfo)	FP Identified (Zaman Yetmemis gibi)
				R	Cast info(similar to writerinfo)	FP Identified (Zaman Yetmemis gibi)
				R	Director info(similar to writerinfo)	FP Identified (Zaman Yetmemis gibi)
				W	Writer info	FP Identified (Zaman Yetmemis gibi)
				W	Producer info	FP Identified (Zaman Yetmemis gibi)
				W	Cast info	FP Identified (Zaman Yetmemis gibi)
				X	Confirm/Error	FP Identified (Zaman Yetmemis gibi)
				X	Person	FP Identified (Zaman Yetmemis gibi)
						1E, 1R, 1X for additional FP: ”List Person” 1E, 5R, 5X for additional FP: ”Retrieve Selected Person”

Table D.12: MM COSMIC FSM Results for Measurer # 12

FUR	**FP#**	**FP Name**	**Fp?**	**DM Type**	**Data Group Desc.**	**Comments**
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	
1	2	List Persons	1	E	List persons request	
				R	Persons info	
				X	Persons info	
1	3	Retrieve Person	1	E	Retrive person details request	
				R	Person details info	
				X	Person details info	
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	
				E	Writer info	Combined as E Movie Info
				E	Producer info	Combined as E Movie Info
				E	Cast info	Combined as E Movie Info
				E	Director info	Combined as E Movie Info
				W	Movie Info	
				W	Writer info	Combined as W Movie Info
				W	Producer info	Combined as W Movie Info
				W	Cast info	Combined as W Movie Info
				W	Director info	Combined as W Movie Info
				X	Error/Confirmation	No X for Error/Confirmation after Add
						1E for request all person list
3	6	Query Movie	1	E	Query Parameters (movie title, year)	
				R	Movie Info	

				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	
3	7	List Movie Details	1	E	Selection of the movie	
				R	Movie Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	Counted once for each type
				X	Movie Info	
				X	Writer info	Combined as X Movie: List of Movies
				X	Producer info	Combined as X Movie: List of Movies
				X	Cast info	Combined as X Movie: List of Movies
				X	Director info	Combined as X Movie: List of Movies
				X	Person info (Name) (counted just once for writer, producer, cast and director)	Combined as X Movie: List of Movies
						1R for Genre Type 5R for Person info (Name) for each person type
4	8	Query Person	1	E	Query Parameters (person name)	Query and Details Listing Combined
				R	Person Info	Query and Details Listing Combined
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	Query and Details Listing Combined
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	
				X	Person Info	
				X	Writer info	Combined as Xperson
				X	Producer info	Combined as Xperson
				X	Cast info	Combined as Xperson

				X	Director info	Combined as Xperson
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	Combined as Xperson
						4R for Movie Info counted for each person type
5	10	List Movies	1	E	Request for a list of movies	
				R	Movie info	
				X	Movie info (title, year)	
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error / Confirmation 1W for Genre
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	From additional FP: ”Check if the person is director” No R to get related data before Delete

				W	Writer info	
				W	Producer info	
				W	Cast info	
				X	Confirm/Error	
				X	Person	No X after Delete - Expected: Return to list as confirmation
						1W Person Info 1W Cast counted twice 1E, 1X for additional FP: ”Check if the person is director”

Table D.13: MM COSMIC FSM Results for Measurer # 13

FUR	**FP#**	**FP Name**	**Fp?**	**DM Type**	**Data Group Desc.**	**Comments**
1	1	Add Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error/Confirmation after Add
1	2	List Persons	1	E	List persons request	
				R	Persons info	
				X	Persons info	
1	3	Retrieve Person	1	E	Retrive person details request	
				R	Person details info	
				X	Person details info	
1	4	Update Person	1	E	Person info	
				W	Person info	
				X	Error/Confirmation	No X for Error / Confirmation after Update
2	5	Add Movie	1	E	Movie Info (title, year, prod.company & genre)	
				R	Person Info (just once for writer, producer, cast and director dropdown lists)	Counted once for each type
				X	Person Info (just once for writer, producer, cast and director dropdown lists)	Counted once for each type
				E	Writer info	Combied as E Movie Details Info
				E	Producer info	Combied as E Movie Details Info

				E	Cast info	
				E	Director info	Combied as E Movie Details Info
				W	Movie Info	
				W	Writer info	
				W	Producer info	
				W	Cast info	
				W	Director info	
				X	Error/Confirmation	
						1R, 1X for Genre 4R, 4X for counting Person info for each type 1E for Movie Details in additional FP: " Save Movie" 1W for Genre 1E Movie Info in additional FP: " Quit Without Saving" 1X Error/Confirmation in additional FP:"Quit Without Saving"
3	6	Query Movie	1	E	Query Parameters (movie title, year)	
				R	Movie Info	
				X	Movie Info (title, year)(OOI:set of movies that match the criteria)	
3	7	List Movie Details	1	E	Selection of the movie	
				R	Movie Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Person info (Name) (counted just once for writer, producer, cast and director)	Counted once for each type
				X	Movie Info	
				X	Writer info	Combined as X Movie: List of Movies
				X	Producer info	Combined as X Movie: List of Movies
				X	Cast info	Combined as X Movie: List of Movies
				X	Director info	Combined as X Movie: List of Movies
				X	Person info (Name) (counted just once for writer, producer, cast and director)	No X due to missing OOI
						1R for Genre 1R for "Lookup Info" 3R for Person info (Name) for each person type 1X for Genre

4	8	Query Person	1	E	Query Parameters (person name)	
				R	Person Info	
				X	Person Info (name)(OOI:Set of persons .t.m.t.c)	
4	9	List Person Details	1	E	Selection of the person	
				R	Person Info	
				R	Writer info	
				R	Producer info	
				R	Cast info	
				R	Director info	
				R	Movie info (counted just once for writer, producer, cast and director)	Counted once for each type
				X	Person Info	
				X	Writer info	
				X	Producer info	
				X	Cast info	
				X	Director info	
				X	Movie info (Title, year) (counted just once for writer, producer, cast and director)	No X due to missing OOI
						4R for Movie Info counted for each person type
5	10	List Movies	1	E	Request for a list of movies	
				R	Movie info	
				X	Movie info (title, year)	
5	11	Delete Movie	1	E	Selection of the movie	
				W	Movie Info	
				R	Writer info (OOI:Set of Writers that are associated with a selected movie)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	No R to get related data before Delete
				W	Writer info	

				W	Producer info	
				W	Cast info	
				W	Director info	
				R	Movie Info	No R after Delete (to be used in Confirmation:X)
				X	Movie info (title, year)	No X to display data as confirmation
						1X for Error / Confirmation 1W for Genre
6	12	Delete Person	1	E	Selection of the person	
				R	Writer info (OOI:Set of movie Writers that are associated with a selected person)	No R to get related data before Delete
				R	Producer info(similar to writerinfo)	No R to get related data before Delete
				R	Cast info(similar to writerinfo)	No R to get related data before Delete
				R	Director info(similar to writerinfo)	
				W	Writer info	
				W	Producer info	
				W	Cast info	
				X	Confirm/Error	
				X	Person	No X after Delete - Expected: Return to list as confirmation
						1W Person Info 1E, 1R, 1X for additional FP: ”List Persons”

APPENDIX E

Appendix E: 4FSM Model for Movie Manager Specifications

Table E.1: 4FSM Model for Movie Manager Application Specifications

Stimulus	**Stimulus FMC**	**Response** **(sender,receiver,cmd)**	**Response FMCs**		
			Category (E,X,C,U)	**FM (default:data group), indicate cmds**	**Sender, Receiver only for X, E**
Add person requested	usr,sw,add user	a person working in movie industry is added	E	Person	usr
			U	Person	
			X	Error/Confirmation	usr
List persons requested	usr,sw,list persons	display list of persons in movie industry	E	Person	usr
			C	Person	
			X	Error/Confirmation	usr
Display person requested	usr,sw,display person	display details of the selected persons	E	Person	usr
			C	Person	
			X	Error/Confirmation	usr
Update person requested	usr,sw,update person	update details of the selected persons	E	Person	usr
			U	Person	
			X	Error/Confirmation	usr
Add Movie F.Form requested	usr,sw,display main movie form	display main movie form and genres	C	Genre	
			X	Genre	usr
Add main movie information requested	usr,sw, display second form	get main movie information and display next form	E	Movie	
			C	Person	

			X	Person	usr
Add movie information requested	usr,sw, save movie information	save all movie information	E	Director	usr
			E	Writer	usr
			U	Movie	
			E	Producer	usr
			E	Cast	usr
			U	Movie	
			U	Director	
			U	Writer	
			U	Producer	
			U	Cast	
			X	Error/Confirmation	usr
Query Movie requested	usr,sw, query movies	display movies that match parameters	E	MovieQuery	usr
			C	Movie	
			X	Movie	usr
Display Movie Details requested	usr,sw, display movie	display selected movie details	E	Movie	usr
			C	Movie	
			C	Director	
			C	Writer	
			C	Producer	
			C	Cast	
			R	Movie	
			X	Movie	usr
			X	DirectorwithName	usr
			X	WriterwithName	usr
			X	ProducerwithName	usr
			X	CastwithName	usr
Query Person requested	usr,sw, query persons	display persons that match the given name	E	PersonNameWord	usr
			C	Person	
			X	Person	usr
Display Person details requested	usr,sw, display person details	display selected person details	C	Director	
			C	Writer	
			C	Producer	
			C	Cast	
			X	Person	usr

			X	MoviesDirected	usr
			X	MoviesWritten	usr
			X	MoviesProduced	usr
			X	MoviesActed	usr
List Movies Requested	usr,sw, display movie	display all movies	C	Movie	
			X	Movie	usr
Delete Movie Requested	usr,sw,delete movie	delete movie	E	Movie	
			C	Director	
			C	Writer	
			C	Producer	
			C	Cast	
			U	Movie	
			U	Director	
			U	Writer	
			U	Producer	
			U	Cast	
			X	Error/Confirmation	usr
			C	Movie	
			X	Movie	usr
Delete Person Requested	usr,sw,delete person	delete person	E	SetofMovies	
			C	Director	
			C	Writer	
			C	Producer	
			C	Cast	
			U	Writer	
			U	Producer	
			U	Cast	
			X	Error/Confirmation	usr

APPENDIX F

Appendix F: Case Study 4-Specifications, Questionnaire and 4FSM Models

Web Ordering System (WOS) Requirements & Specifications

B2B Product Order System. XYZ Company gets orders from commercial customers that sell products in their stores. XYZ desires to have an online Web Ordering System (WOS) to improve business performance by automation of order reception, preparation and payment processes.

Environment and Business Requirements. Each XYZ customer mostly orders products that have similar properties. For this purpose XYZ maintains a catalog of generic product properties (i.e keywords, tags). Each customer's preferred set of generic product properties is already known such that they will be able see a list of products that match the properties when they access the system. Shipping costs are charged to customers. For cost minimization, the customers desire to give multiple-product orders and they desire to prepare an order in the system incrementally and at different times before submitting it. Orders can be in one status: open, in-progress and sent to delivery. Each customer can have one open order at most at a time. Customers have prepaid deposits at their XYZ accounts. While checking out their order, they want to select whether they want to pay from prepaid deposit or online credit card payment. For deposit payments customers should provide PaymentKey information which is generated by a key generator device each customer owns. For payment validation purposes the key should be kept with time stamp information in XYZ system. The payment validation is performed via another Finance Software System (FinSys).Credit card payments need to be performed by using an online secure bank web service.

Upon the submission of orders, XYZ warehouse packing machine prepares an empty package with the orderID barcode printed on it and customer receives order in progress e-mail. The warehouse workers collect the ordered products and place them in the package. Before loading the completed orders on the delivery trucks, the orderID bar codes are scanned to inform the system that it has been shipped. XYZ supervisor can request a report of orders created in a time period for planning and monitoring purposes. The supervisor needs the report for many reasons such as:

- asking warehouse workers to get prepared for potential orders or prioritize order preparations
- replying customer inquiries about orders and payments
- calling customers for inventory status before they submit their orders

Software Requirement Specifications

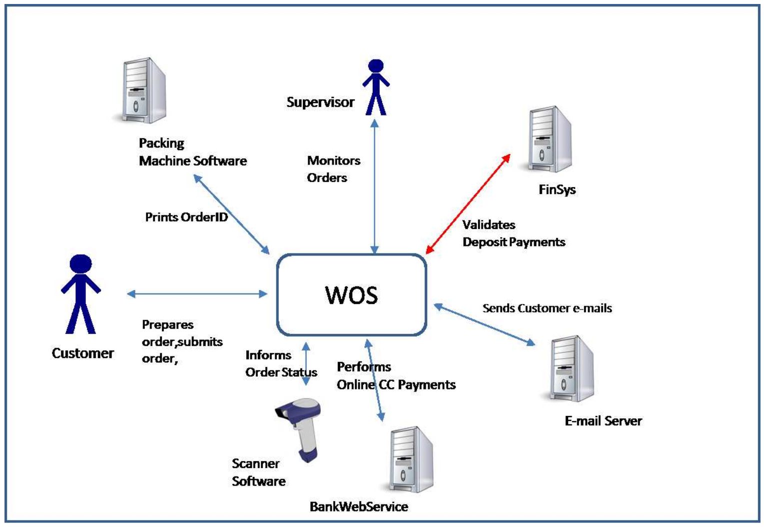


Figure F.1: WOS Software Functional Users

- Customers should login WOS by typing their names and passwords. If an attempt to login fails an error message is shown, o.w. after login verification WOS checks if there is an open (unsubmitted) order for the customer. If there is, WOS displays the open order details (orderID, order creation date) and the products that match customer preferences listed below a product query form. o.w, it creates a new open order and displays the same details. The query form comes with one search field, one combo-box that includes list of generic product property labels.

- When the customer user types in the text in the field and "submits" a query, WOS returns the list of products (productno,price,title) that contain the query text in product title. If the user selects an property label from the combo-box then the search text is ignored and the query returns the products that possess the selected property. Customer preferences are ignored in queries.

- Customer adds an item to the order by entering a quantity value and requesting "add item" for a selected product. WOS takes productid and indicated quantity from the interface and adds an item to the order and displays a success message.

- When the customer clicks "view order", the system takes orderid from the interface and views order details (orderid, creationdate, total cost) and order-item details (product title, quantity, unitprice, itemcost).

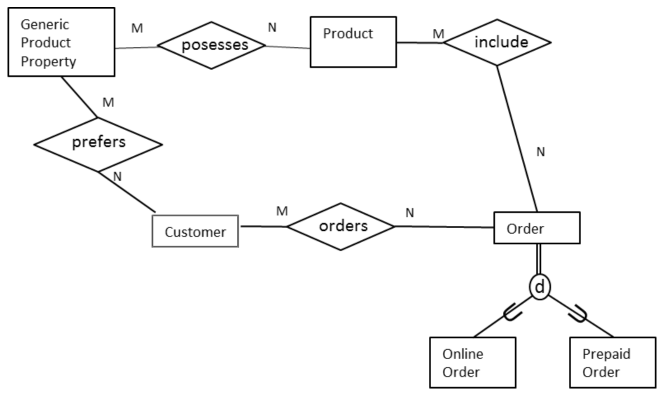
- For the deletion of an order item customer first selects "view order" then from the order details view, customer selects "delete" for an item and WOS takes the productno and orderid from the interface and deletes the item from the order and WOS displays the updated order details (orderid, creationdate, total cost) and order-item details (product title, quantity, unitprice, itemcost).

- Customer cancels an order by requesting "cancel" from order main page. WOS first shows a confirmation dialog, when user clicks "ok" it takes orderid from the interface deletes all order related information and returns a success message.

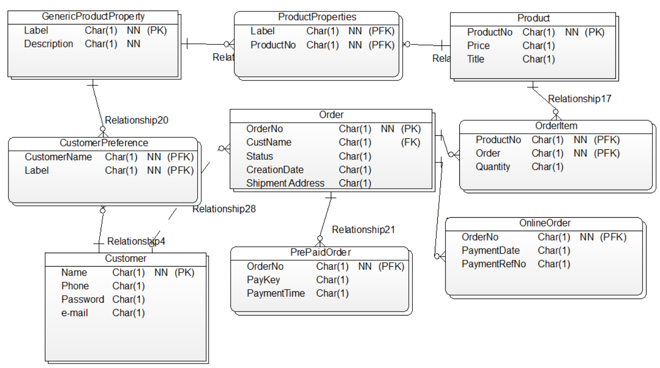
- In order to check out and submit the order, the customer first clicks "checkout" button. WOS displays checkout wizard. In the first wizard form, user selects from two options: (1) Pay from Deposit, (2) Pay Online and enters a PaymentKey (1) or CCNumber (2) into a field. In the "next" form, user enters shipping address. When the user requests "submit order" if the payment type is selected as "from deposit" the payment key and timestamp information is saved for later validation. If credit card option is selected WOS sends a credit card payment request to bank web service. Web service returns a PaymentReferenceNo as a confirmation. In both cases system saves address information and updates order status to ”In progress”. WOS also sends orderID to XYZ packing machine software and sends a confirmation e-mail message to customer including orderID and in-progress status.

- When Warehouse worker scans the OrderID printed on the order package, scanner software sends orderID to WOS and WOS updates order status to ”Sent to Delivery”.

- XYZ supervisor enters two dates (from and to) for a report of orders created in the specified time period. Report includes all orders details (see sample report at the end).



(a) E-R Diagram



(b) 3NF Relations

Figure F.2: WOS Entities and Relations

Web Ordering System (WOS) 4FSM MODEL

Table F.1: Reference 4FSM Model for WOS Specifications

Stimulus	**Stimulus FMC** (sender,receiver,cmd)	**Response**	**Response FMCs**		
			(E,X,C,U)	**FM**(indicate cmds)	**Sender, Receiver for X, E**
login requested	cust,sw,login	check login , error or page,display order	E	Customer	customer
			C	Customer	
			X	loginerr	customer
			C	CustomerPreference	
			C	order	
			U	order	
			X	order	customer
			C	GenericProduct Property	
			C	Product Property	
			X	GenericProduct Property	
			C	Product	
			X	Product	
query requested	cust,sw,query products	display query results	E	Product	customer
			E	Product Property	customer
			C	Product	
			C	Product Property	
			X	Product	customer
add item requested	cust,sw,add item request		E	OrderItem	customer
			U	OrderItem	
			X	Success	customer
view order requested	cust,sw,view order request	view order details	E	order	
			C	Order	
			C	Orderitem	
			C	Product	
			X	Orderitemdetails	customer
			X	Orderdetails	customer
delete item requested	cust,sw,delete item req	delete item and show	E	Orderitem	customer
			U	Orderitem	
			C	Order	
			C	Orderitem	
			C	Product	

			X	Orderitem	customer
			X	Orderdetails	customer
cancel order request	cust,sw,click cancel req	click cancel	n.a.	interface mechanics	n.a
Cancel Confirmed	cust,sw, confirmation	delete all order information	E	Order	customer
			C	Orderitem	
			U	Order	
			U	Orderitem	
			X	Success	customer
checkout requested	cust,sw,click checkout	show wizard	n.a.	interface mechanics	n.a
fill payment requested	cust,sw,click next	show address	E	PrepaidOrder (key)	customer
			E	OnlineOrder (cc number)	customer
submit requested	cust,sw,click submit	record payment	E	Order	customer
			U	PrepaidOrder	
			U	OnlineOrder	
			U	Order	
			X	withdraw:cmd	Bank WS
			X	OnlineOrder (cc number)	Bank WS
			E	OnlineOrder (refno)	Bank WS
			X	PrintBarcode:cmd	PackMac
			X	Order	PackMac
			X	Orderwithe-mail	MailServer
			X	send e-mail:cmd	MailServer
			C	Customer	
order scanneed	scanner,sw,scan	update status	E	Order	customer
			U	Order	
report requested	supervisor,sw,report request	display report	E	OrderReportInterval	supervisor
			C	Order	
			C	OrderItem	
			C	Customer	
			C	OnlineOrder	
			C	PrepaidOrder	
			C	Product	
			X	All Customer Order	supervisor
			X	Orderdetails	supervisor
			X	Orderitemdetails	supervisor

4FSM Model Evaluation Questionnaire

For each of the following paired statements, please mark a cross over the box, which most closely matches your opinion. Give your honest opinion, based on your experience using the 4FSM model and the construction procedure.

4FSM model data concepts are intuitive						4FSM model data concepts are abstract
4FSM model behavioral concepts are intuitive						4FSM model behavioral concepts are abstract
4FSM model behavioral concepts are easy to understand						4FSM model behavioral concepts are hard to understand
4FSM model data concepts are easy to understand						4FSM model data concepts are hard to understand
4FSM model construction procedure was easy to follow and apply						4FSM model construction procedure was hard to follow and apply
Overall, 4FSM model construction was easy						Overall, 4FSM model construction was challanging
All 4FSM rules guides me identifying the model concepts						All 4FSM rules do NOT guide me identifying the model concepts
4FSM model provides means for verification of a constructed model						4FSM model do NOT provide means for verification of a constructed model
It is easy to learn 4FSM model data and behavioral concepts						It is difficult to learn 4FSM model data and behavioral concepts
Training and documentation was sufficient to apply the model accurately						Training and documentation was NOT sufficient to apply the accurately
The case specifications were simpler than the previous sw I have measured						The case specifications were harder than sw I have measured
The case specification was simpler than Movie Manager from an FSM point of view.						The case specification was harder than Movie Manager from an FSM point of view.
The case specifications were closer to real life projects compared to the sw I have measured						The case specifications were NOT close to real life projects compared to the sw I have measured
4FSM concepts are concrete and solid. Once I learn it, I can easily extend the FSM definitions and rules to any SRS and obtain consistent results.						4FSM concepts are subjective. I can achieve consistency by memorizing the examples for each requirement in order to obtain consistent results.

Please provide free text answers to each of the following questions:

1. What do you think about the required skills to understand and apply 4FSM? Compare it to the skills for FSM method you know?

2. What are the most significant contributions of 4FSM model you think with respect to model construction challenges you have met?

3. Do you have any suggestions for making 4FSM model easier to apply?

Please write any other comments you would like to make about the FSM method in the space below.

Web Ordering System 4FSM Models Constructed by Measurers

Table F.2: WOS Models Constructed by Measurer 1

Stimulus	**Stimulus FMC** (sender,receiver,cmd)	**Response**	**Response FMCs**		
			(E,X,C,U)	**FM**(indicate cmds)	**Sender, Receiver for X, E**
login requested	display open orders	cust,sw,login	E	Customer	customer
			C	Customer	
			X	success/error	
			C	order	
			X	order	
			C	CustomerPreference	
			C	ProductProperty	
			C	Product	
			C	GenericProduct Property	
			C	Product Property	
			X	GenericProduct Property	
query submitted	display query results	cust,sw,query products	E	Product	customer
			E	Product Property	
			C	Product	
			C	Product Property	
			X	Product	customer
add an item requested	place order	cust,sw,add item request	E	OrderItem	
			E	Product	
			U	OrderItem	
			X	Success	
view order requested	view order details	cust,sw,view order request	e	order	
			C	order	
			C	orderitem	
			C	product	
			X	orderdescription	
			X	ordersummary	
delete item requested	delete item and show	cust,sw,delete item req	E	orderdescription	
			U	orderitem	
			C	order	
			C	orderitem	
			X	orderdescription	
			X	ordersummary	

cancel order request	cancel order	cust,sw, confirmation	E	order	
			C	orderitem	
			U	order	
			U	orderitem	
			X	success/error	
payment requested	show address	cust,sw,click next	E	Payment Data	customer
			E	Order	
			U	Order	
			U	PrepaidOrder	
			X	Payment data	Bank WS
			E	BankPaymentData	Bank WS
			U	OnlineOrder	
			X	Order	PackMac
			C	Customer	
			X	Orderwithe-mail	MailServer
			X	Customer	
			X	PrintBarcode:cmd	PackMac
			X	e-mail:cmd	MailServer
			X	send e-mail:cmd	MailServer
order scanned	update status	scanner,sw,scan	E	Order	customer
			U	Order	
report requested	display report	supervisor,sw,report request	E	ReportDate	supervisor
			C	Product	
			C	OrderItem	
			C	Order	
			C	OnlineOrder	
			C	PrepaidOrder	
			X	RManagerSummary	
			X	ROrderSummary	
			X	RItemSummary	

Table F.3: WOS Models Constructed by Measurer 2

Stimulus	**Stimulus FMC** (sender,receiver,cmd)	**Response**	**Response FMCs**		
			(E,X,C,U)	**FM**(indicate cmds)	**Sender, Receiver for X, E**
login requested	cust,sw,login	login to system	E	Customer	customer
			C	Customer	
			C	order	
			X	order	
			C	Product	
			X	Product	
			U	order	

			X	loginerr	
			C	CustomerPreference	
			C	Product Property	
			C	GenericProduct Property	
			X	GenericProduct Property	
search product button clicked	cust,sw,query products	display related products	E	Product	customer
			C	Product	
			C	Product Property	
			X	Product	
			E	GenericProduct Property	
add item requested	cust,sw,add item request	idem item to order	E	OrderItem	customer
			U	OrderItem	
view order requested	cust,sw,view order request	display order details	E	Order	customer
			C	order	
			C	orderitem	
			X	ordertotalcost	
			X	orderdetails	
			C	Product	
delete item requested	cust,sw,delete item req	delete an order item	E	orderitem	customer
			E	Order	
			U	orderitem	
			C	order	
			C	Product	
			X	orderitemdetails	
			X	orderdetails	
Order cancel requested	cust,sw, confirmation	order cancelled	E	order	customer
			C	orderitem	
			U	order	
			U	orderitem	
			X	success	
submit order requested	cust,sw,click submit	record payment	E	PrepaidOrder	customer
			E	OnlineOrder	
			U	PrepaidOrder	
			U	OnlineOrder	
			U	Order	
			C	Customer	

			X	withdraw:cmd	Bank WS
			E	OnlineOrder	Bank WS
			X	Order	PackMac
			X	Orderwithe-mail	MailServer
			X	CreditCardNo	Bank WS
			E	Order	customer
order scanneed	scanner,sw,scan	update status	E	Order	Scanner Soft
			U	Order	
report requested	supervisor,sw,report request	display report	E	OrderReportInterval	supervisor
			C	Order	
			C	OrderItem	
			C	Customer	
			C	OnlineOrder	
			C	PrepaidOrder	
			C	Product	
			X	All Customer Order	
			X	Orderdetails	
			X	orderitemdetails	

Table F.4: WOS Models Constructed by Measurer 3

Stimulus	**Stimulus FMC** (sender,receiver,cmd)	**Response**	**Response FMCs**		
			(E,X,C,U)	**FM**(indicate cmds)	**Sender, Receiver for X, E**
login requested	cust,sw,login	check login , error or page,display order	E	Customer	customer
			C	Customer	
			X	loginerr	
			C	order	
			U	order	
			X	order	
			C	CustomerPreference	
			C	Product Property	
			C	GenericProduct Property	
			X	GenericProduct Property	
			C	Product	
			X	Product	
query requested	cust,sw,query products	show search results	E	Product	customer
			E	Product Property	

			C	Product	
			C	Product Property	
			X	Product	
add item requested	cust,sw,add item request		E	OrderItem	customer
			U	OrderItem	
			X	Success	
view order requested	cust,sw,view order request	view order details	E	Order	customer
			C	Product	
			C	order	
			C	orderitem	
			X	orderitemdetails	
			X	orderwithTotalCost	
delete orderitem requested	cust,sw,delete item req	delete item	E	orderitem	
			U	orderitem	
			C	Product	
			C	order	
			C	orderitem	
			X	orderitemdetails	
			X	orderdetails	
Cancel Order clicked	cust,sw, confirmation	delete all order information	E	order	customer
			C	orderitem	
			U	order	
			U	orderitem	
			X	success	
checkout clicked	cust,sw,request		E	PrepaidOrder	customer
			E	OnlineOrder	
			E	Order	
			U	PrepaidOrder	
			U	OnlineOrder	
			U	Order	
			X	withdraw:cmd	Bank WS
			X	CreditCardNo	Bank WS
			E	PaymentReferenceNo	Bank WS
			X	PrintBarcode:cmd	PackMac
			X	Order	PackMac
			C	Customer	
			X	send e-mail:cmd	MailServer
			X	Orderwithe-mail	MailServer
			X	Customer	MailServer
order scanneed	scanner,sw,scan	update status	E	Order	Scanner sw
			U	Order	

report requested	supervisor,sw,report request	display report	E	OrderReportInterval	customer
			C	Order	
			C	OrderItem	
			C	Customer	
			C	OnlineOrder	
			C	PrepaidOrder	
			C	Product	
			X	All Customer Order	
			X	Orderdetails	
			X	orderitemdetails	

Table F.5: WOS Models Constructed by Measurer 4

Stimulus	**Stimulus FMC** (sender,receiver,cmd)	**Response**	**Response FMCs**		
			(E,X,C,U)	**FM**(indicate cmds)	**Sender, Receiver for X, E**
login requested	cust,sw,login	check login	E X	Customer loginerr	customer
			C	Customer	
			C	order	
			X	order	
			C	Customer Property	
			C	GenericProduct Property	
			X	Product	
			U	order	
			C	Product Property	
			C	Product	
			X	GenericProduct Property	
Submit text query			E	Product	customer
			C	Product	
			X	Product	
Select by label	cust,sw,query products		E	Property	customer
			C	Product Property	
			C	Product	
			X	Product	
add item requested	cust,sw,add item request		E	OrderItem	customer
			U	OrderItem	
			X	Success	

view order requested	cust,sw,view order request		C	order	customer
			E	order	
			C	orderitem	
			C	Product	
			X	orderitemdetails	
			X	orderdetails	
delete item requested	cust,sw,delete item req		E	Product	customer
			E	Order	
			U	orderitem	
			C	Product	
			C	orderitem	
			X	orderitemdetails	
			X	orderdetails	
Cancel Order	cust,sw, confirmation		E	order	customer
			X	Confirmation	
			E	order	
			U	order	
			C	orderitem	
			U	orderitem	
			X	error/success	
click checkout deposit	cust,sw,click checkout		E	Paymentkey	customer
			E	Shipmentaddress	
			U	PrepaidOrder	
			U	Order	
			X	Order	
			X	Orderwithe-mail	
click checkout creditcard			E	Creditcard	customer
			E	Shipmentaddress	
			X	PaymentRequest	Bank WS
			E	Payment Reference	Bank WS
			U	Order	
			X	Order	PackMac
			X	Orderwithe-mail	
			X	PaymentRequest:cmd	Bank WS
order scanneed	scanner,sw,scan		E	Order	Scanner sw
			U	Order	
report requested	supervisor,sw,report request		E	OrderCreationFrom	supervisor
			E	OrderCreationTo	

			C	Order	
			C	OrderItem	
			C	Customer	
			C	OnlineOrder	
			C	PrepaidOrder	
			C	Product	
			X	OrderReport	
			X	Order_Customer	
			X	Order_Items	
			X	Order_creation	

Table F.6: WOS Models Constructed by Measurer 5

Stimulus	**Stimulus FMC** (sender,receiver,cmd)	**Response**	**Response FMCs**		
			(E,X,C,U)	**FM**(indicate cmds)	**Sender, Receiver for X, E**
login requested	cust,sw,login	check login	E	Customer	customer
			X	loginerr	
			C	Customer	
			C	order	
			X	order	
			C	Customer Property	
			C	GenericProduct Property	
			X	Product	
			U	order	
			C	Product Property	
			C	Product	
			X	GenericProduct Property	
Submit text query			E	Product	customer
			C	Product	
			X	Product	
Select by label	cust,sw,query products		E	Property	customer
			C	Product Property	
			C	Product	
			X	Product	
add item requested	cust,sw,add item request		E	OrderItem	customer
			U	OrderItem	
			X	Success	
view order requested	cust,sw,view order request		C	order	customer

			E	order	
			C	orderitem	
			C	Product	
			X	orderitemdetails	
			X	orderdetails	
delete item requested	cust,sw,delete item req		E	Product	customer
			E	Order	
			U	orderitem	
			C	Product	
			C	orderitem	
			X	orderitemdetails	
			X	orderdetails	
Cancel Order	cust,sw, confirma-tion		E	order	customer
			X	Confirmation	
			E	order	
			U	order	
			C	orderitem	
			U	orderitem	
			X	error/success	
click checkout deposit	cust,sw,click check-out		E	Paymentkey	customer
			E	Shipmentaddress	
			U	PrepaidOrder	
			U	Order	
			X	Order	
			X	Orderwithe-mail	
click checkout credit-card			E	Creditcard	customer
			E	Shipmentaddress	
			X	PaymentRequest	Bank WS
			E	Payment Refer-ence	Bank WS
			U	Order	
			X	Order	PackMac
			X	Orderwithe-mail	
			X	PaymentRequest:cmd	Bank WS
order scanneed	scanner,sw,scan		E	Order	Scanner sw
			U	Order	
report requested	supervisor,sw,report request		E	OrderCreationFrom	supervisor
			E	OrderCreationTo	
			C	Order	
			C	OrderItem	

			C	Customer	
			C	OnlineOrder	
			C	PrepaidOrder	
			C	Product	
			X	OrderReport	
			X	Order_Customer	
			X	Order_Items	
			X	Order_creation	

Table F.7: WOS Models Constructed by Measurer 6

Stimulus	**Stimulus FMC** (sender,receiver,cmd)	**Response**	**Response FMCs**		
			(E,X,C,U)	**FM**(indicate cmds)	**Sender, Receiver for X, E**
login requested	cust,sw,login	verify,show order	E	Customer	customer
			C	Customer	
			X	error	
			C	order	
			X	order	
			C	CustomerPreference	
			C	Product Property	
			C	Product	
			U	order	
			X	Product	
add item requested	cust,sw,add item request		E	OrderItem	customer
			U	OrderItem	
			X	Success	
search requested	search	display query results	E	Product	customer
			E	Product Property	
			C	Product	
			C	Product Property	
			X	Product	
view order requested	cust,sw,view order request	view order details	C	order	customer
			E	Orderid	
			C	orderitem	
			C	Product	
			X	orderitemdetails	
			X	orderdetails	
delete item requested	cust,sw,delete item req	delete item	E	orderitem	customer
			C	orderitem	
			U	orderitem	

			C	Product	
			R	Product	
			X	orderdetails	
cancel order request	cust,sw,click cancel req	cancel order	E	order	customer
			C	orderitem	
			U	order	
			U	orderitem	
			X	success	
checkout requested	cust,sw,click check-out	submit order	E	PrepaidOrder	customer
			E	CreditCardNo	
			E	Order	
			U	PrepaidOrder	
			X	withdraw:cmd	Bank WS
			X	CreditCardNo	Bank WS
			E	OnlineOrder	Bank WS
			U	OnlineOrder	
			U	Order	
			X	PrintBarcode:cmd	PackMac
			X	Order	PackMac
			X	Order	Customer
			X	Orderwithe-mail	MailServer
			X	send e-mail:cmd	MailServer
order scanneed	scanner,sw,scan	update status	E	Order	Scanner sw
			U	Order	
report requested	supervisor,sw,report request	display report	E	OrderReportInterval	supervisor
			C	Order	
			C	OrderItem	
			C	Customer	
			C	Product	
			X	All Customer Order	
			X	Orderdetails	
			X	orderitemdetails	

Table F.8: WOS Models Constructed by Measurer 7

Stimulus	**Stimulus FMC** (sender,receiver,cmd)	**Response**	**Response FMCs**		
			(E,X,C,U)	**FM**(indicate cmds)	**Sender, Receiver for X, E**
login requested	cust,sw,login	create,check order,display	E	Customer	customer

			C	Customer	
			X	loginerr	
			C	order	
			U	order	
			C	CustomerPreference	
			C	GenericProduct Property	
			C	Product Property	
			X	Product	
			C	Product	
			X	GenericProduct Property	
			X	order	
search product requested	cust,sw,query products	display query results	E	Product	customer
			E	Product Property	
			C	Product	
			C	Product Property	
			X	Product	
add item requested	cust,sw,add item request		E	OrderItem	customer
			U	OrderItem	
			X	Success	
view order requested	cust,sw,view order request	view order details	C	product	customer
			E	Order	
			C	orderitem	
			X	orderitemdetails	
			X	orderdetails	
delete item requested	cust,sw,delete item req	delete item and show	E	orderitem	customer
			U	orderitem	
			X	orderitemdetails	
			X	orderdetails	
cancel order request	cust,sw,click cancel req	click cancel	n.a.		
Cancel Confirmed	cust,sw, confirmation	delete all order information	E	order	customer
			C	orderitem	
			U	order	
			U	orderitem	
			X	success	
checkout requested	cust,sw,click checkout	show wizard	n.a.		
fill payment (next)	cust,sw,click next	show address	E	PrepaidOrder	customer

			E	OnlineOrder	
submit order requested	cust,sw,click submit	record payment	E	Order	customer
			U	PrepaidOrder	
			X	withdraw:cmd	Bank WS
			X	CreditCardNo	Bank WS
			E	OnlineOrder	Bank WS
			U	Order	
			X	PrintBarcode:cmd	PackMac
			X	Order	PackMac
			X	send e-mail:cmd	MailServer
			X	Orderwithe-mail	MailServer
			U	OnlineOrder	
			X	Customer	MailServer
order scanneed	scanner,sw,scan	update status	E	Order	Scanner sw
			U	Order	
report requested	supervisor,sw,report request	display report	E	OrderReportInterval	supervisor
			C	Order	
			C	OrderItem	
			C	Customer	
			C	OnlineOrder	
			C	PrepaidOrder	
			C	Product	
			X	All Customer Order	
			X	Orderdetails	
			X	orderitemdetails	

Table F.9: WOS Models Constructed by Measurer 8

Stimulus	**Stimulus FMC** (sender,receiver,cmd)	**Response**	**Response FMCs**		
			(E,X,C,U)	**FM**(indicate cmds)	**Sender, Receiver for X, E**
login requested	cust,sw,login	process login	E	Customer	customer
			X	loginerr	
			C	order	
			C	CustomerPreference	
			C	Product	
			C	GenericProduct Property	
			X	order	
			X	GenericProduct Property	

			x	product	
query submitted	cust,sw,query products	query products	E	Product	customer
			C	Product	
			X	Product	
			E	GenericProduct Property	
			C	GenericProduct Property	
add item requested	cust,sw,add item request		E	OrderItem	customer
			C	Product	
			C	Order	
			U	OrderItem	
			X	Success	
view order requested	cust,sw,view order request	view order details	E	order	customer
			C	order	
			C	orderitem	
			C	Product	
			X	order_totalcost	
			X	product_orderitem	
delete item requested	cust,sw,delete item req	delete item from an order	E	Product	customer
			E	Order	
			U	OrderItem	
			U	Order_totalcost	
			U	product_orderitem	
			X	order_totalcost	
			X	product_orderitem	
Cancel Confirmed	cust,sw, confirmation	cancel order	E	order	customer
			X	error/confirmation	
			E	error/confirmation	
			U	order	
			C	Product	
			U	orderitem	
			X	error/confirmation	
checkout requested	cust,sw,click next	checkout order	E	Order	customer
			E	PrepaidOrder	
			E	CreditCardNo	
			C	PrepaidOrder	
			X	withdraw:cmd	Bank WS
			X	CreditCardNo	Bank WS
			U	Order	

			U	OnlineOrder	
			U	PrepaidOrder	
			X	Orderwithe-mail	MailServer
			X	Order	PackMac
			X	PrintBarcode:cmd	PackMac
			X	send e-mail:cmd	MailServer
order scanneed	scanner,sw,scan	update status	E	Order	Scanner sw
			U	Order	
report requested	supervisor,sw,report request	display report	E	fromTodates	supervisor
			C	Customer	
			C	Order	
			C	Product	
			C	OrderItem	
			X	Product_Quantity	
			X	Order_total	
			X	All Customer Order	
			C	OnlineOrder	
			C	PrepaidOrder	

Table F.10: Ratings for Questions in 4FSM Evaluation Questionnaire

		Measurers							
		1	2	3	4	5	6	7	8
Questions	Q1	2	2	2	4	2	2	5	1
	Q2	2	2	2	4	1	1	5	2
	Q3	2	2	1	2	1	2	1	1
	Q4	1	2	1	2	2	1	1	1
	Q5	2	2	1	1	1	1	1	1
	Q6	2	2	1	1	1	2	1	2
	Q7	2	1	1	2	3	2	1	2
	Q8	2	1	1	2	2	2	3	3
	Q9	1	2	1	1	2	2	1	1
	Q10	1	2	2	3	1	1	1	1
	Q11	3	1	4	1	3	3	3	2
	Q12	3	1	4	2	2	3	3	5
	Q13	2	4	1	2	4	2	1	1
	Q14	2	3	1	2	1	2	1	2

VITA

Barış Özkan received his bachelor degree in Industrial Engineering from Middle East Technical University (METU) in 2000. In 2006, he had his M.Sc. degree in Software Management from Informatics Institute, METU.

During 2000 and 2008, he worked as business analyst, software developer, software architect and IT consultant in various ICT product development and process improvement projects in public and private sectors.

Since 2008, he has participated in a number of research projects and assisted courses related to software engineering in particular software measurement and estimation. In the field of software size measurement, estimation and benchmarking, he participated in two research projects entitled ”Unified Software Estimation Method and Toolset” and ”Software Benchmark Dataset for Estimation and a Process Oriented Estimation Method” supported by The Scientific and Technological Research Council of Turkey(TUBITAK).

His research interests include; software measurement, process improvement, business process modeling, requirements engineering and empirical software engineering.

ODTÜ
ENFORMATİK ENSTİTÜSÜ

YAZARIN

Soyadı : ÖZKAN

Adı : BARIŞ

Bölümü : BİLİŞİM SİSTEMLERİ

TEZİN ADI (İngilizce) : A SOFTWARE FUNCTIONALITY MODEL FOR FUNCTIONAL SIZE MEASUREMENT

TEZİN TÜRÜ : Yüksek Lisans .... Doktora: X

1) Tezimden fotokopi yapılmasına izin vermiyorum ☐

2) Tezimden dipnot gösterilmek şartıyla bir bölümünün fotokopisi alınabilir ☐

3) Kaynak gösterilmak şartıyla tezimin tamamının fotokopisi alınabilir ☐

Yazarın imzası .... Tarih ....