AN AUTOMATED QUALITY MEASUREMENT APPROACH FOR BUSINESS PROCESS MODELS

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

BY

ÖZGE GÜRBÜZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

SEPTEMBER 2011

Approval of the Graduate School of Informatics

Prof. Dr. Nazife BAYKAL
Director

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

Prof. Dr. Yasemin YARDIMCI ÇETİN
Head of Department

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

Dr. A. Selçuk GÜCEĞLİOĞLU	Assoc. Prof. Dr. Onur DEMİRÖRS
Co-Advisor	Supervisor

Examining Committee Members

Prof. Dr. Semih BİLGEN	(METU, EE)	_____________________
Assoc. Prof. Dr. Onur DEMİRÖRS	(METU, IS)	_____________________
Dr. A. Selcuk GÜCEĞLİOĞLU	(BAŞBAKANLIK)	_____________________
Assist. Prof. Dr. Aysu BETİN CAN	(METU, IS)	_____________________
Assoc. Prof. Dr. Altan KOÇYİĞİT	(METU, IS)	_____________________

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 wok.

Name, Last name: Özge Gürbüz

Signature : _________________

ABSTRACT

AN AUTOMATED QUALITY MEASUREMENT APPROACH FOR BUSINESS PROCESS MODELS

Gürbüz, Özge

Ms.c., Department of Information Systems
Supervisor: Assoc. Prof. Dr. Onur Demirörs
Co-Advisor: Dr. A. Selçuk Güceğlioğlu

September 2011, 103 Pages

Business process modeling has become a common need for organizations. Therefore process quality is also having an important role for the organizations. The most of the quality studies are based on cost and time which can be analyzed during or after the execution of the business processes. There are also quality measures which help analyzing measures before the execution of the business processes. This type of measures can give early feedback about the processes. There are three frameworks defined in the literature for a more comprehensive measurement. One of the frameworks is adapted from software programs and it aims to enable process design to be less error-prone, understandable and maintainable. The second framework is adapted from object-oriented software designs and it provides object-oriented view to the design of the business process. The last framework is adapted from ISO/IEC Software Product Quality enabling to measure the quality of process itself rather than the design. By conducting a case study, the measures defined in the frameworks are explored in terms of applicability, automation potential and required time and effort on a set of business process model. As a result of this study it is observed that measurement takes time and requires effort and is always error-prone. Therefore, an approach is implemented by automating the measures which have automation potential, in order to decrease the required time and effort and also to increase the accuracy of the measurement. The second case study is then conducted on a set of another business process models in order to validate the approach.

Keywords: Business Process Quality, Software Quality, Quality Metrics

ÖZ

İŞ SÜREÇ MODELLERİ İÇİN OTOMATİKLEŞTİRİLMİŞ KALİTE ÖLÇÜM YAKLAŞIMI

Gürbüz, Özge
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Danışmanı: Doc. Dr. Onur Demirörs
Yardımcı Danışman: Dr. A. Selçuk Güceğlioğlu

Eylül 2011, 103 Sayfa

İş süreç modelleme, organizasyonlar için yaygın bir ihtiyaç haline gelmiştir. Bu yüzden de süreçlerin kalitesi de organizasyonlar için önemli bir yer teşkil etmektedir. Kalite çalışmalarının çoğu maliyet ve zaman üzerine kurulmuştur ve bunlar ancak süreçler uygulanırken veya uygulandıktan sonra analiz edilebilirler. Aynı zamanda iş süreçlerinin uygulanmasından önce de analiz edilebilirliği sağlayan kalite ölçümleri bulunmaktadır. Bu tip ölçümler süreçler hakkında erken geri bildirim verebilmektedirler. Literatürde daha kapsamlı bir ölçüm için tanımlanmış üç çerçeve vardır. Çerçevelerden bir tanesi yazılım programlarından geçirilmiştir ve süreç tasarımın daha az hataya açık, anlaşılır ve bakımın kolay yapılabilir olmasını sağlamaktadır. İkinci çerçeve objeye-dayalı yazılım tasarımından uyarlanmıştır ve iş süreçlerinin tasarımına objeye-dayalı bakış sağlamaktadır. Son çerçeve ISO/IEC Yazılım Ürün Kalitesi'nden geçirilmiş olup tasarımdan ziyade süreçlerin kendilerinin kalitesini ölçmeyi mümkün kılmaktadır. Yapılan bir durum çalışmasıyla, çerçevelerdeki ölçümler iş süreç modelleri seti üzerinde, uygulanılabilirlik, otomatikleştirme potansiyeli ve harcanan zaman ve efor açısından araştırılmıştır. Bu çalışma sonucunda ölçümlerin zaman aldığı, efor gerektirdiği ve hataya açık olduğu gözlemlenmiştir. Bu yüzden otomatikleştirme potansiyeli olan ölçümler otomatikleştirilerek, harcanan zaman ve eforu azaltmak ve aynı zamanda ölçümlerin doğruluğunu artırmak için bir yaklaşım geliştirilmiştir. Sonrasında başka bir iş süreç kümesinin üzerinde yaklaşımı doğrulamak için ikinci bir durum çalışması yapılmıştır.

Anahtar Kelimeler: İş Süreç Kalitesi, Yazılım Kalitesi, Kalite Ölçüleri

ACKNOWLEDGEMENT

I would like to thank the people in my life that have guided me when I needed direction, gave me encouragement when I was having doubts, motivated me when I started feeling overwhelmed, supported me when I needed support, and were patient with me when I had so many questions.

Let me begin by expressing my sincere appreciation to my advisor, Assoc. Prof. Dr. Onur Demirörs, and my co-advisor, Dr. A.Selçuk Güceğlioğlu, for their continuous guidance, support, patience and encouragements throughout my study.

I am grateful for my father Abdullah Gürbüz, my mother Sema Gürbüz and my sister Ayten Gürbüz for their endless patience, encouragement and support throughout my educational pursuit. I am really happy and fortunate to be your daughter.

I would also want to thank Ayşegül Özkaya for being there with me on every important moment of my life. She and Esin Karabacakoğlu's support in this study means a lot for me.

My destiny friends Serhat Peker, Davut Çavdar, Gökçen Yılmaz and M. Erhan Uyar made the time writing and researching for this thesis enjoyable. I can never forget their motivations. Thank you for your endless support, encouragement and patience.

Duygu Fındık, Nurcan Alkış, Elif Aydın and Zeynep Başgöze are my angels. I owe them so much for their Turkish coffees, fortunes, encouragement and suggestions since the beginning of this study.

I would also like to thank Tom Boulton for his support on this study. I could never forget those articles.

Also I am thankful to the staff of Informatics Institute for their helps in every stage of the bureaucratic tasks.

I want to thank to The Scientific and Technical Research Council of Turkey (TÜBİTAK) for supporting me with scholarship during my MSc study.

This thesis is dedicated to my mom, dad and sister...

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

During the last decade business process modeling has become a common need for a number of organizations. Business process models are used for a variety of purposes, including, but not limited to, establishing an execution consistency, optimization, automation, measurement and certification. Several notational conventions can be used for business process models. Some examples of the modeling notations are: Business Process Modeling Notation (BPMN), Petri nets, activity diagrams and (Event-Driven Process Chain) EPC diagrams.

Business process modeling is a technique for business quality management. Most of the studies in the literature concentrate on the cost and time effects, return investment and market share attributes of the process quality. However, these attributes can only be determined during or after the execution of the business processes and are named as "*post-execute*". An obvious disadvantage of the post-execute attributes is the need to utilize the resources that can be avoided. In other words, by post execute measures process improvement opportunities cannot be predicted in advanced. On the other hand, there are quality measures that concentrate on coupling, cohesion and complexity of the business process design and on maintainability, reliability, functionality and usability of the business processes. These quality attributes give important feedback about the design of the business process model and the identification of the processes. These attributes can be determined before the business process execution and are named as "*pre-execute*".

The quality attributes defined for measuring the business processes' design are mostly adapted from software domain. There is a significant similarity between a process and software. Both of them have inputs, activities/functions, and outputs. Because of this similarity many researchers have adapted coupling, cohesion, complexity and size measures from software programs (Vanderfeesten 2007). A similar adaptation is performed based on object-oriented software design since business process modeling notations are similar to software design notations (Khlif 2009). Maintainability, reliability, functionality and usability attributes are adapted from ISO 9126 Software Product Quality (Guceglioglu 2006). These measures are developed both as singular measures and as frameworks containing a set of measures. These measures can be used before the execution of the processes and therefore have the potential to provide early feedback to the organization. And enable to measure the business processes according to their design, element relations within a process and the quality of the process itself.

1.1 Problem Statement

It is obvious that the quality improvement initiatives would be more valuable before executing the process designed. Therefore our study focus the pre-execution quality attributes. Most of the singular measures are developed to measure the complexity attribute of the business process models. For more comprehensive measurement needs, researchers have also defined quality measures as frameworks. Vanderfeesten's quality framework is adapted from software programs and includes coupling, cohesion, complexity, and size measures (Vanderfeesten 2007). Khlif's quality model is adapted from object-oriented designs and includes coupling and cohesion measures (Khlif 2009). Gucegliogluʼs pre-enactment model is adapted from ISO 9126 Software Product Quality and includes maintainability, reliability, functionality and usability measures (Guceglioglu 2006).

Experimental studies show that quality measurement frameworks provide necessary feedback for common process problems (Mendling 2006, Vanderfeesten 2007). However, as each measure requires analyzing and counting different process components and these calculations require significant effort, time and human judgment, and the task become subjective and costly. Besides time and subjectivity considerations, manual calculation is error-prone.

In this study we developed an automation approach for Vanderfeesten's, Khlif's and Gucegliogluʼs measures and apply the approach for Human Resources Management processes, which include 21 processes (Gurbuz 2011). Applying the quality measures on a single business process set requires 13 hours and there are many other processes in the organization. Also since the process improvement is a continuous work, the quality of the measures has to be measured continuously. This means that the required time and effort for quality measurement become significant

As far as known from the literature review, only Vanderfeesten implemented plug-in for the ProM framework. With this plug-in it is possible to automatically measure control flow complexity, weighted coupling, density and size of EPC process model (Vanderfeesten 2007). The business processes can be analyzed with respect to complexity, coupling and size views. Although these measures give insights about the process's mode, there are still two other frameworks for measuring the quality of business processes.

1.2 Approach

The aim of this thesis is to reduce time and effort required for quality measurement as well as to provide more accurate results by automating process quality measurement. A case study is performed to explore the available measures in the literature. Vanderfeesten's, Khlif's and Guceglioglu's measures are applied on a set of Human Resources Management processes manually. This application revealed the measures' applicability on EPC diagrams, automation potential and required time and effort. As a result, the business quality measures that are applicable using EPC diagrams are identified. From the applicable measures, the measures which do not need human interpretation and have automation potential as well as the requirements for automation are determined. Based on the requirements defined during the first case study, an Automated Quality Measurement (AQM) tool was developed as an add-on to the COSMOS Tool. The COSMOS Tool is a Meta-Model Editor integrated into the KAMA conceptual modeling environment. COSMOS is developed as a project in Bilgi Grubu (Bilgi Grubu Ltd. Şti 2008). With the AQM integrated in to the COSMOS, designers are able to measure the quality of the business processes while designing them.

AQM is then validated by applying the measures on a set of Supply Chain Management processes. The application results are compared in terms of time and effort requirements and accuracy of the calculations.

1.3 Outline

The thesis has been organized into five chapters. The first chapter is the introduction and includes the overview of the study, problem statement and approach. The second chapter is the related research which gives information about the available quality measures in the literature.

The third chapter describes the approach by first introducing the Kama Tool and how it works. Then it gives details about the automation tool and describes how the selected measures are implemented. The fourth chapter explains the exploration and validation case studies. The last chapter concludes the thesis with conclusion and future work part.

CHAPTER 2

BACKGROUND AND RELATED RESEARCH

Business process improvement initiatives have a significant impact on organizations (Ralph 2007). Process baselines including process models are frequently established during these initiatives. Process baselines enable measuring process quality from various perspectives. Several research studies were focused on complexity of business processes (Cardosa 2006, Gruhn 2006, Ghani 2008, Muketha 2010). All these researches points out the similarity between software programs and business processes. Because of this similarity most of the measures are adapted from software measures. These include Number of Activities which is adapted from Lines of Code (Azuma 1994), control flow complexity which is adapted from McCabe's Cyclomatic Complexity (McCabe 1976), information flow metric adapted from Henry and Kafura (Henry 1981) and Process Quality Measurement Model (Guceglioglu 2011).

Besides complexity measures defined in the literature, other singular measures such as density (Mendling 2006), weighted coupling (Vanderfeesten 2007) and cross-connectivity (Vanderfeesten 2008) measures were also defined in this research area. To provide more comprehensive approach, researchers also established frameworks for process quality. Vanderfeesten's framework of quality inspires from software programs and relates 5 design principles; coupling, cohesion, complexity, modularity and size (Vanderfeesten 2006). This model aims to enable the process design to be less error-prone, easier to understand and easier to maintain. A cooperative work is Khlif's framework. Khlif's measures are adapted from object-oriented (OO) software design and categorized in two classes; coupling and cohesion (Khlif 2009). One different view he points out in his measures is that this cohesion measures the cohesion between the process tasks. The last framework is Guceglioglu's pre-enactment model

(Guceglioglu 2006). This model is not really about how the processes are designed but how qualified they are from a different perspective. As it is indicated by the name in this framework the process is analyzed as a product.

This chapter is organized as follows; in the following sections Vanderfeesten's Quality framework will be introduced, in the third section Khlif's object-oriented adapted quality measures will be introduced, in the fourth section Guceglioglu's Pre-Enactment Model will be introduced and in the fifth section Vanderfeesten's singular measure on cross-connectivity will be introduced. Lastly this chapter will be concluded by selecting the quality measures from all of the three frameworks which can be automated.

2.1 Vanderfeesten Framework of Quality

This framework is developed on 5 measures; coupling, cohesion, complexity, modularity and size. Vanderfeesten marks 2 coupling measures in their framework. The first one is the density measure, which is actually defined for to measure complexity, and the second one is the weighted coupling measure (Vanderfeesten 2007). Vanderfeesten references a cohesion measure defined for workflow processes. For the complexity measure Vanderfeesten marked on Control-Flow Complexity measure and for modularity they denoted that they haven't met any modularity measure in their research. Lastly for size number of functions, events, ORs, XORs and ANDs are calculated.

2.1.1 Coupling

Coupling measures the degree of connections between activities in the process model.

Density. Vanderfesssten references Mendling's density measure. This measure is defined for measuring the complexity of EPC business process model (Mendling 2006). Mendling indicates that there are many complexity measures adapted from software world but it is still hard to contrast the complexity of the models which are in different size. But because coupling measures the interconnections between nodes it is also related to the degree and density of measures.

Density measure formulas are defined as follows;

$$a_{min\,=n-1.}\ (1)$$

$$c_{maxeven} = (c/_2 + 1)^2. \ (2)$$

$$c_{maxodd} = (\frac{c-1}{2} + 1)^2 + \frac{c-1}{2} + 1. \ (3)$$

$$c_{c \leq 1} = 1. \ (4)$$

$$d_{even} = \frac{a - a_{min}}{c_{maxeven} + 2*(e+f) - a_{min}}. \ (5)$$

$$d_{odd} = \frac{a - a_{min}}{c_{maxodd} + 2*(e+f) - a_{min}}. \ (6)$$

$$d_{c \leq 1} = 1. \ (7)$$

n=number of nodes, a=number of arcs, c=number of connectors, e= number of events, f= number of functions.

Weighted Coupling Metric. Vanderfeesten defines this measure for measuring the coupling according to the tasks' connection (Vanderfeesten 2007). He believes that this measure can be helpful when evaluating ease of understanding of the process. This measure as shown in the formula calculates the coupling value differently for AND, XOR and OR connectors. They tested this measure on SAP reference EPC model with simple size measures to see if it can really predict errors. According to tests they had 2 issues; one is that it really has positive impact on error probability but on the other hand density and size were not enough to explain variance of errors (Vanderfeesten 2007).

Weighted coupling is defined as follows;

$$CP = \frac{\sum_{t_1, t_2 \in T} connected \ (t_1, t_2)}{|T| * (|T| - 1)}. \ (8)$$

Where $connected(t_1, t_2) =$

$$\left\{ \begin{array}{ll} 1 & , if \ (t_1 \rightarrow t_2) \ and \ (t_1 \neq t_2) \\ 1 & , if \ (t_1 \rightarrow AND \rightarrow t_2) \ and \ (t_1 \neq t_2) \\ \frac{1}{(2^m - 1)*(2^n - 1)} + \frac{(2^m - 1)*(2^n - 1) - 1}{(2^m - 1)*(2^n - 1)} * \frac{1}{m*n} & , if \ (t_1 \rightarrow OR \rightarrow t_2) \ and \ (t_1 \neq t_2) \\ \frac{1}{m*n} & , if \ (t_1 \rightarrow XOR \rightarrow t_2) \ and \ (t_1 \neq t_2) \\ 0 & , if \ (t_1 = t_2) \end{array} \right\}. \ (9)$$

t_1 and t_2 are the activities, m is the number of ingoing arcs to the connector and n is the outgoing arcs from the connector.

2.1.2 Complexity

Control-Flow Complexity. Control-Flow Complexity measure is adapted from McCabe's cyclomatic number (Cardoso 2006). This measure differs from McCabe's measure in the way that every node in business process can have different meaning. For as; OR split is different than XOR and AND split. Formulas are as follows;

$$CFC_{XOR}(a) = n. \quad (10)$$

$$CFC_{OR}(a) = 2^n - 1. \quad (11)$$

$$CFC_{AND}(a) = 1. \quad (12)$$

$$CFC(P) = \sum_{a \in P, a \text{ is a } XOR-split} CFC_{XOR}(a) + \sum_{a \in P, a \text{ is a } OR-split} CFC_{OR}(a) + \sum_{a \in P, a \text{ is } AND-split} CFC_{AND}(a). \quad (13)$$

This measure is evaluated in terms of Weyuker's properties and it can be used for examining complexity of processes (Muketha 2011).

2.1.3 Modularity

As far as known from the literature there is no measure developed for modularity yet. Modularity measures the degree of separated parts of a module. Low modularity ends with more errors, but also high modularity is not desirable.

2.1.4 Size

This measure is used to measure the length of a process. Vanderfeesten counts number of events (NOE), functions (NOF), XORs, ANDs and ORs (Vanderfeesten). It is thought that when number of related issues is high than complexity gets higher, ease of understandability gets lower and can result in mistakes.

2.2 KHLIF's Framework of Quality

Khlif introduces new coupling and cohesion measures adapted from object oriented software measures. They believe that there are similarities between object oriented (OO) software and Business Process Modeling Notation (BPMN) (Khlif 2009). Khlif compares the object oriented software and BPMN notation as statically; class-process, method-task, variable-data object, comment line-annotation, interface of a class-interface of a process, local data in a class-process tasks data objects, data used by a class-data object used by process tasks. Dynamically method invocation correspond reception or message flow by a task (Khlif 2009).

2.2.1 Coupling Metrics

Imported and Exported Coupling. In software engineering domain Imported Coupling calculates the number of class that is being used by each class C. In other words the adapted version for business processes, Imported Coupling of a Process (ICP*), counts for each process the number of flows sent by itself* (Khlif 2009).

On the other hand Exported Coupling calculates how many other class uses a class C; which in the business process terms *counts the number of flows received* (Khlif 2009) that they name as Exported coupling of a Process (ECP).

They noted that a process with a high value of ICP means that it depends highly on the other processes. In addition they indicate that this may cause to high costs and error probabilities also increase in delays. For a high value of EPC they believe that it has influence on whole model which can cause problem since every process in the model will depend on their incoming flows.

Response for Process Coupling. Response for Process (RFP) coupling is adapted from responses to class coupling, which actually examines control flows in the name of coupling (Khlif 2009).

In the following formula; *RS* is the set of all responses of a process, is the set of tasks invoked by a task *i* in the process and is the set of all tasks *j* in the process.

$$RFP=|RS|. \quad (14)$$

$$RS = \{T_j\} \cup \{R_i\}. \quad (15)$$

Khlif notes that the higher value of RFP means higher complexity of a process. They argue this claim as that if a process can invoke a larger number of other tasks or a process this means that it is complex so needs high understanding.

Locality of Data-Based Coupling. This measure is for calculating the data that the process uses over the total data that it uses or produces. DT_i ($1 \leq i \leq n$) is the set of data associated to task T_i within the activity and L_i ($1 \leq i \leq n$) is the set of data produced by other activities and used by a task $T_i$the activity (Khlif 2009).

$$LDA = \frac{\sum_{i=1}^{n} |L_i|}{\sum_{i=1}^{n} |DT_i|}. \quad (16)$$

For example for an activity A, input is a list *L* (information carrier in our model) which is produced by another activity and end of that activity output is another list *L2* (information carrier) so locality of data activity (LDA), which is the adapted name, $\{"L"\}/\{"L","L2"\}$ = ½.

Khlif claimed that high value of LDA means more adapted to reuse and easier to test than those which has low LDA.

2.2.2 Cohesion Metrics

Tight Process Cohesion. Originally tight class cohesion measure counts the percentage of method pairs that are connected directly. Two methods were directly related in the case of using directly or indirectly same variable (Khlif 2009). Adapted version of this measure is called Tight Process Cohesion (TPC) which is calculated as follows;

$$TPC = \frac{NSPDC}{NSP}. \quad (17)$$

$$NSP = \frac{[N*(N-1)]}{2}. \quad (18)$$

NSP is the maximum number of public task pairs and N is the number of tasks in a measured process. On the other hand NSPDC is the number of direct connections between its public tasks. A task can be directly connected by another task if they use same data directly or indirectly. Data is used directly if it is produced by the task that it uses, or it is used indirectly if a task T receives it directly or indirectly a sequence flow from the task T.

Khlif notes that 0 TPC value means that the tasks within the process are not directly related and it is the worst cohesion scenario.

Loose Process cohesion. Unlike the TPC metric Loose Process Cohesion (LPC) metric calculates the percentage of task pairs either directly or indirectly related (Khlif 2009). NSPC is the number of direct or indirect connections between the tasks of the measured process. Formula is as follows;

$$LPC = \frac{NSPC}{NSP}. \text{ (19)}$$

It is denoted that like TPC metric high value of LPC is best quality scenario.

2.3 Guceglioglu's Pre-Enactment Model

This model is adapted from ISO/IEC 9126 Software Product Quality (Guceglioglu 2006). Guceglioglu draws attention on the similar logical structure between a process and software. It is indicated that both a process and software has inputs, activities/functions, and outputs (Guceglioglu 2005). With this model it is believed that organizations can achieve pre-execute results about the process. This means that executing the processes and finding out errors afterwards will end up with a higher cost and time spent. With pre-execute model, quality can be understood earlier, before execution.

Guceglioglu defines this model in four-leveled structure like in ISO/IEC 9126 Software Product Quality. The first level is category level and it is Quality. The second level is characteristics and includes Maintainability, Reliability, Functionality and Usability. The third level is sub-characteristics of the second level and defines Analyzability, Fault Tolerance, Recoverability, Suitability, IT Based Functionality, Accuracy, Interoperability, Security, Understandability, Learnability, Operability, Attractiveness metrics. Finally, the last level defines the attributes of these sub-characteristics which are cited in the following sections.

2.3.1 Analyzability Metrics

Complexity (CX). This measure calculates the number of decision points over total number of activities. Formula is as follows;

$$CX = 1 - \frac{A}{B}. \ (20)$$

For overall evaluation, A is the number of connectors and B is the number of activities. In detail each decision type is counted separately such as for structured, unstructured, and semi-structured decisions. Structured decision is defined as programmable decision, unstructured decisions needs creative decision, and semi-structured decision may be repetitive but can also require some human intuition (Guceglioglu 2006). The higher value of CX means better analyzability.

Coupling. Coupling (CP) measure counts the interactions of the process with other processes. This measure is for the processes which have interactions with other processes. Formula is as follows;

$$CP = 1 - \frac{A}{B}. \ (21)$$

A is the number of interactions and B is the number of activities. Higher the value of CP better the analyzability is.

2.3.2 Fault Tolerance Metrics

Failure Avoidance. Failure Avoidance (FA) measure is for identifying the activities in which review, inspection, checkpoint or similar techniques are applied. So in this way with this measure user-based mistake are tried to be minimized. Formula is as follows;

$$FA = \frac{A}{B}. \ (22)$$

A is the number activities in which review, inspection, checkpoints are applied and B is the number of activities. High value of FA means better failure avoidance.

2.3.3 Recoverability Metrics

Restorability. Restorability (R) measures how completely are the activities is recorded on paper or on computers. Formula is as follows;

$$R = \frac{A}{B}. \ (23)$$

A is the number of recorded activities and B is the number of activities. Higher value of R indicates better restorability of a process.

Restoration Effectiveness. Restoration Effectiveness (RE) is to identify effectiveness of restoration. This metric aims to calculate which recorded activities' can be saved in case of lost. Formula is as follows;

$$RE = \frac{A}{B}. \ (24)$$

A is the number of activities which can be restored and B is the number of recorded activities. Higher value of RE means better restoration effectiveness.

2.3.4 Suitability Metrics

Functional Adequacy. Functional Adequacy (FAD) is for identifying adequacies of the process activities in practice. In other words with this measure it would be possible to define if there is unconformity between the activity in the practice and activity defined in related documents. Formula is as follows;

$$FAD = \frac{A}{B}. \ (25)$$

A is the number of adequate activities with their definitions in regulatory documents and B is the number of activities. Higher value of FAD is better functional adequacy.

Functional Completeness. Functional Completeness (FC) identifies if there are missing activities in process according to regulatory documents. Formula is as follows;

$$FC = 1 - \frac{A}{B}. \ (26)$$

A is the number of activities which are defined in the regulatory documents but forgotten in process design and B is the number of activities. High value of FC indicates better functional completeness.

2.3.5 IT Based Functionality Metrics

IT Usage. Guceglioglu defines IT Usage (ITU) for measuring the IT usage in activities. Formula is as follows;

$$ITU = \frac{A}{B}. \quad (27)$$

A is the number of activities in which IT applications are used for creating, deleting, updating or searching purposes and B is the number of activities. High value of ITU means high IT usage.

IT Density. Guceglioglu defines IT Density (ITD) for specifying the use of IT applications. This measure is calculated as;

$$ITD = \frac{A}{B}. \quad (28)$$

A is the number of forms, reports or other documents which are prepared, updated, deleted or searched by using IT applications and B is the number of forms, documents in the process. High value of ITD is more IT density.

2.3.6 Accuracy Metrics

Computational Accuracy. Computational Accuracy (CA) measures the implementation of the accuracy requirements in process design. Formula is as follows;

$$CA = \frac{A}{B}. \quad (29)$$

A is the number of activities in which specific accuracy requirements have been implemented as defined in regulatory document and B is the number of activities which have accuracy requirements. High value of CA is more accurate of a process.

2.3.7 Interoperability Metrics

Data Exchangeability. Data Exchangeability (DE) measure specifies the operations applied to the data received from other process. By this measure it can be seen that if the input data of an activity had operation before used and can be compared by the number of activities interactions with other processes. Formula is as follows;

$$DE = \frac{A}{B}. (30)$$

A is the number of activities in which no change is performed on the received data before using it and B is the number of activities which have interactions with other processes. Higher value of DE, more data exchangeability it is.

2.3.8 Security Metrics

Access Auditability. Access Auditability (AA) measure is defined for auditing access to process activities so that who accessed to the data can be analyzed. Formula is as follows;

$$AA = \frac{A}{B}. (31)$$

A is the number of activities which have access to data and this access can be audited with its actor and B is the number of activities which have access to the data sources. High value of AA is more auditable.

2.3.9 Usability Metrics

Functional Understandability. Functional Understandability (FU) measure is for specifying difficulties for understanding activities. Formula is as follows;

$$FU = \frac{A}{B}. (32)$$

A is the number activities in which staff do not face any difficulties in understanding the tasks and B is the number of activities. High value of FU is better understandability.

2.3.10 Learnability Metrics

Existence in Documents. Existence in Documents (EID) measure is for analyzing the available documents about the model and measures which of the activities are defined in them. Formula is as follows;

$$EID = \frac{A}{B}. \ (33)$$

A is the number of activities which are described in the available documents and B is the number of activities. High value of EID indicates more complete documentation.

2.3.11 Operability Metrics

Input Validity Checking. Input Validity Checking (IVC) measure is for identifying validity checking possibilities for input parameters in the process activities. Formula is as follows;

$$IVC = \frac{A}{B}. \ (34)$$

A is the number of activities in which validity checking can be performed for input parameters and B is the number of activities. High value of IVS is better input validity checking.

Undoability. This measure is for identifying activities which can be undone. Formula is as follows;

$$U = \frac{A}{B}. \ (35)$$

A is the number of activities which can be undone and B is the number of activities. The closer value to 1 means better undoability.

2.3.12 Attractiveness Metrics

Attractive Interaction. Attractive Interaction (AI) measure is for specifying the difficulties or easiness in preparation, deletion or updating forms, reports or other documents used in the activity. Formula is as follows;

$$AI = \frac{A}{B}. \ (36)$$

A is the number activities which staff doesn't face any difficulties and B is the number of recorded activities. High value of AI is more attractive interaction.

2.3 Vanderfeesten's Cross Connectivity Measure

This measure is defined for measuring how strongly the model is connected (Vanderfeesten 2008).

Cross-connectivity measure is defined in 5 steps;

1. *Weight of a Node.* A node can be a task T or a connector C and total nodes N is union of T and C. d is the number of ingoing and outgoing arcs.

$$w(n) = \begin{cases} 1, if\ n \in C\ and\ n\ is\ a\ type\ of\ AND \\ \frac{1}{d}, if\ n \in C\ and\ n\ is\ a\ type\ of\ XOR \\ \frac{1}{2^d-1} + \frac{2^d-2}{2^d-1} * \frac{1}{d}, if\ n \in C\ and\ n\ is\ a\ type\ of\ OR \\ 1 \qquad , \ if\ n \in T \end{cases}. \ (37)$$

2. *Weight of an Arc.* Each arc *a* has a source node (src(a)) and a destination node (dest(a)).

$$W(a) = w(src(a)) * w(dest(a)). \ (38)$$

3. *Value of a Path.* A path p is the sequence of arcs that should be followed between nodes n_1, n_2 : $p = <a_1, a_2, \ldots, a_x>$ and calculated as follows;

$$v(p) = W(a_1) * W(a_2) * \ldots * W(a_x). \ (39)$$

4. *Value of a Connection.* If P_{n_1,n_2} is set of paths between nodes n_1, n_2, value of connection is the maximum value of these paths as follows;

$$V(n_1, n_2) = max_{p \in P_{n_1,n_2}} v(p). \ (40)$$

5. Cross-Connectivity

$$CC = \frac{\sum_{n_1, n_2 \in N} V(n_1, n_2)}{|N| * (|N| - 1)}. \ (41)$$

It is specified that this measure is validated by a set of 12 process models with 25 tasks in each and questionnaire of 73 students, with the goal of evaluating their understandability (Vanderfeesten 2008). And according to the result of this measure, high value means easier understanding of model and low error probability. But it is also noted that this measure alone is not very powerful for determining the understandability of the model but it can be helpful when combined with other existing metrics (Vanderfeesten 2008).

CHAPTER 3

APPROACH: AUTOMATED BUSINESS PROCESS QUALITY MEASUREMENT

This chapter will be introduced in two sections. The first section includes introduction about the environment in which the approach is implemented, the COSMOS Tool, its features and the local database design. The second section introduces the automated quality measurement tool and its implementation.

3.1 The COSMOS: Meta-Model Editor for Conceptual Modeling

3.1.1 Overview

The COSMOS tool is developed as a Meta-Model Editor which is integrated in to the KAMA modeling environment, in a scope of finishing project in Software Management graduate program in Middle East Technical University. This tool provides users to define Meta-Model Entities such as the Elements, Relation Types, Diagram Types and Meta-Models. This way different Domain-Specific Models can be created using defined Meta-Models in the scope of one modeling environment. The tool is supported by a local database to save the models (Khalikov 2008).

High level use case diagram for the COSMOS tool is given in Figure 1.

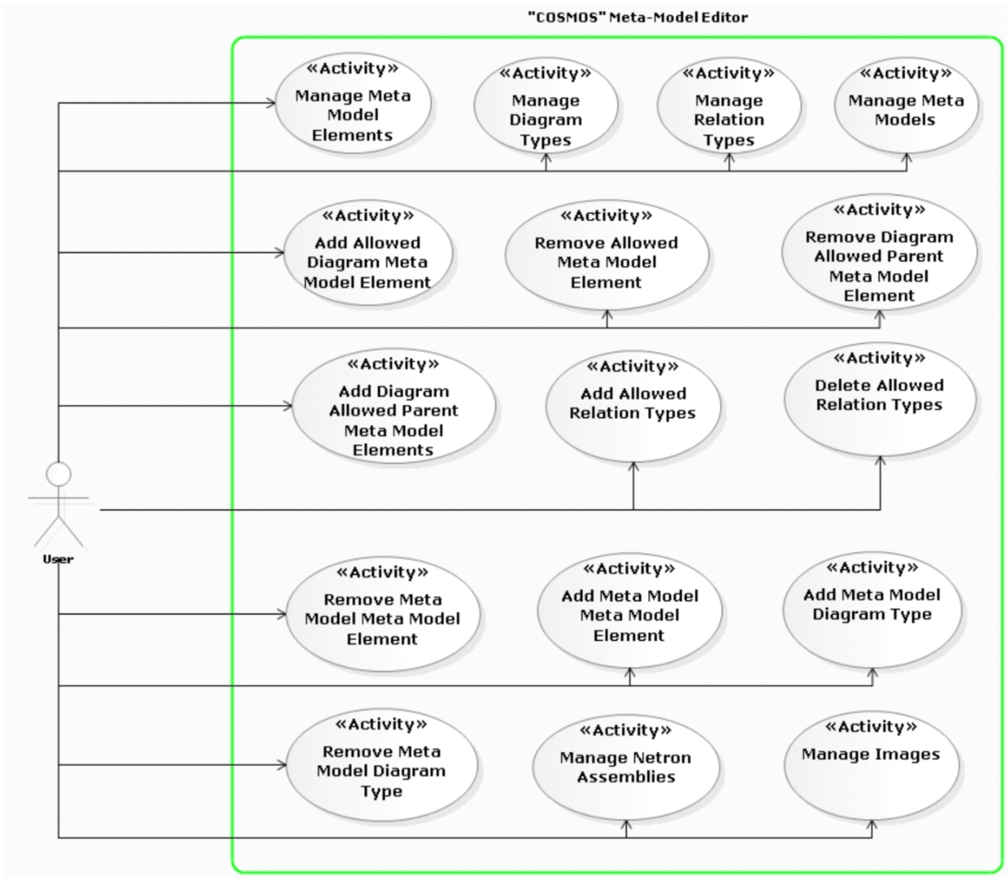


Figure 1. High Level Use Case Diagram of COSMOS

The most significant feature of this modeling environment is that you can create new meta-models, diagram types, and within the new diagram types you can create meta-model elements and manage their relationships. Thus, this modeling environment does not only support one modeling notation. The user can create and enhance a new modeling notation in this tool. The menu bar from COSMOS tool and one example management form are given in the Figure 2 and Figure 3. You can create the Meta Model by typing the name and choosing a picture to represent it. Then you can choose from the existing meta-model elements (or create it by Meta Model Elements Management), and add to the list. You can also choose several diagram types for the created meta-model. Each diagram type has allowed model elements; you can arrange this from Diagram Types Management menu. Relation Types Management enables the users to choose which relations can be done between elements.

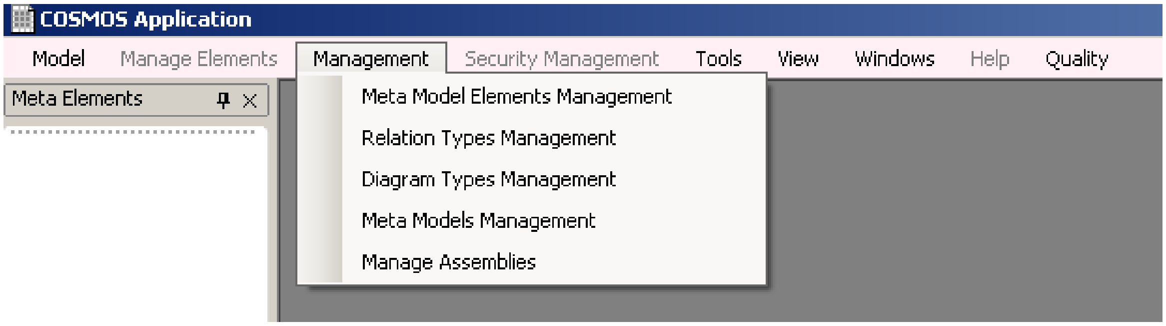


Figure 2. Menu Bar of COSMOS

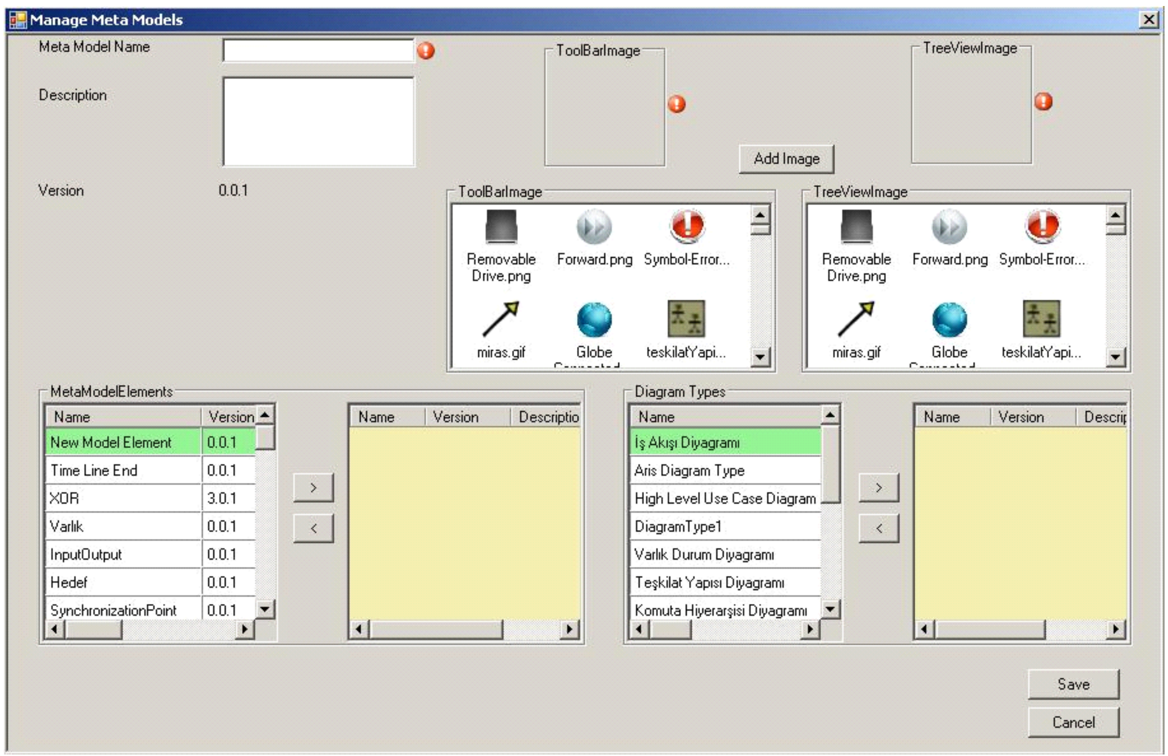


Figure 3. COSMOS Manage Meta-Models

This thesis' approach only concentrates on eEPC modeled business processes. Thus, following sections will discuss mostly on this notation, and the COSMOS's meta-model elements, relation types and database design.

3.1.2 Meta Model Elements

For our study extended Event-driven Process Chain diagram is used. Event-driven Process Chain

(EPC) is a method developed by Scheer, Keller and Nüttgens for the companies to model, analyze and redesign their business processes as the start point of the information systems implementation (Ferdian 2001). EPC is developed within a framework called Architecture of Integrated Information System (ARIS). The COSMOS Tool is developed as in the same approach of the ARIS for modeling EPC diagrams which also has a local database and provides adding new design model elements. EPC diagrams, however, includes events, functions, control flows, process interfaces and connectors. Extended-EPC (eEPC), besides EPC notations, includes input-outputs as information carriers, roles, clusters, telephone and ext.

The notations used in eEPC diagram form are summarized in Table 1.

Table 1. Meta Model Elements in the COSMOS Tool

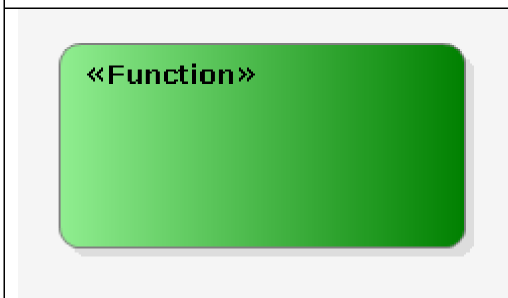
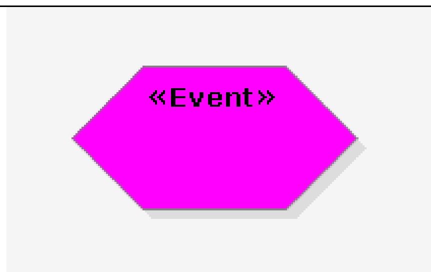
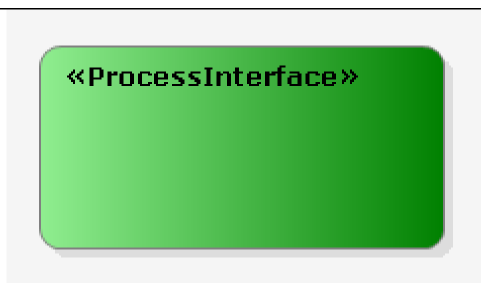
META MODEL ELEMENTS	DESCRIPTIONS
	"Function" shows the activities in the organization
	"Event" describes under what conditions does functions work
	"ProcessInterface" shows other processes used in the process.

Table 1 (cont.)

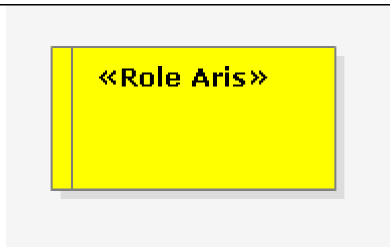
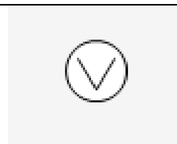
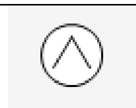
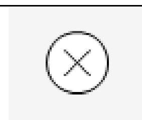
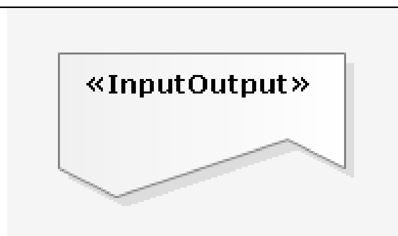
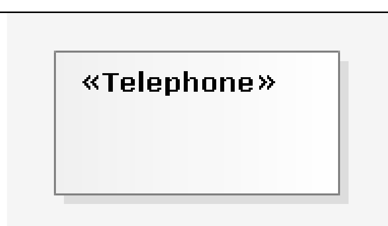
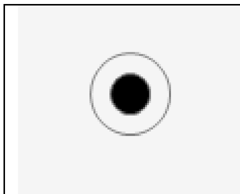
	The role icon indicates the people or organization that takes the role when processing the activity.
	From all of the activities that are connected to OR connector, one or more activities are executed in parallel.
	All of the activities that are connected to AND should be executed in parallel.
	From all of the activities that are connected to XOR connector, only one activity is executed.
	Information carriers can be input or output of functions and portray the objects in the real world, such as forms, materials and documents.
	"Cluster" indicates group of similar things in a particular place, for as in this model the cluster is the database.
	The telephone icon indicates usage of telephone when processing the activity.

Table 1 (cont.)

	The end icon indicates the end of the process model.

3.1.3 Relation Types

Two types of relations are used; “Rect Relation” and “Rect Relation Arrow”. Model Elements that use “Rect Relation” is given in Table 2. Model Elements that use “Rect Relation Arrow” is given in Table 3.

Table 2. . “Rect Relation” Supported Model Elements

Function	—	Role Aris
Process Interface	—	Role Aris
Function	—	Cluster
Process Interface	—	Cluster
Function	—	END icon
Process Interface	—	END icon

Table 3. “Rect Relation Arrow” Supported Model Elements

Function	→	Event
Event	→	Function
Function	→	InputOutput
InputOutput	→	Function
ProcessInterface	→	InputOutput
InputOutput	→	Process Interface
Process Interface	→	Event
Event	→	Process Interface
Function	→	Telephone
Telephone	→	Function

3.1.4 Database Design

Once the new model is created, the *ID* of the model, *Name* of the model, *Creator*, *CreationID* and other necessary information is recorded in the database.

Models

Column Name	Data Type	Length	Allow Nulls
pkey	int	4	
ID	uniqueidentifie	16	
MetaModelID	uniqueidentifie	16	
Name	varchar	50	
Version	varchar	50	
CreationDate	datetime	8	
UpdateDate	timestamp	8	
ApproveDate	datetime	8	✓
PreApprovalDate	datetime	8	✓
CreatorID	uniqueidentifie	16	
PrivacyLevelID	uniqueidentifie	16	
Objective	varchar	4000	✓
Scope	varchar	4000	✓

Figure 4. Models Table

Upon the creation of the model, you should create a diagram in which the diagram will be designed. For as Aris Meta Model includes Aris Diagram Type which supports eEPC diagram type. For this choice in Diagram Table new record with a unique ID for the diagram is created. In this Table, *ModelID* matches with the *ID* in Models Table.

Diagrams

Column Name	Data Type	Length	Allow Nulls
pkey	int	4	
ID	uniqueidentifie	16	
DiagramTypeID	uniqueidentifie	16	
ModelID	uniqueidentifie	16	
ParentID	uniqueidentifie	16	
Name	varchar	255	
CreationDate	datetime	8	
UpdateDate	timestamp	8	
ApproveDate	datetime	8	✓
PreApprovalDate	datetime	8	✓
Version	varchar	50	
CreatorID	uniqueidentifie	16	
PrivacyLevelID	uniqueidentifie	16	

Figure 5. Diagrams Table

After creating the model, model elements can be added. Each model element has a unique id. Once you add the element to your model new id is created in the database.

ModelElements

	Column Name	Data Type	Length	Allow Nulls
🔑	pkey	int	4	
	ID	uniqueidentifie	16	
	MetaModelElementID	uniqueidentifie	16	
	ModelID	uniqueidentifie	16	
	ParentID	uniqueidentifie	16	
	Name	varchar	4000	✓
	Version	varchar	50	
	CreationDate	datetime	8	
	UpdateDate	timestamp	8	
	ApproveDate	datetime	8	✓
	PreApprovalDate	datetime	8	✓
	CreatorID	uniqueidentifie	16	
	PrivacyLevelID	uniqueidentifie	16	
	ContactPoint	varchar	4000	✓
	FidelityAndSecurity	varchar	4000	✓
	Description	varchar	4000	✓

Figure 6. ModelElements Table

As shown in Figure 6, *ID* is unique for each created element. *MetaModelElementID* is dependent on the type of element created. For as, all functions have the same *MetaModelElementID*. *ModelID* represents the model that it is created in, which matches to the *ID* in the Models Table.

The model element created on the diagram is recorded in the *DiagramsToModelElements* Table. *DiagramID* matches with the *ID* in the Diagrams Table. A unique *ID* is created for the Model Elements in the related diagram.

DiagramsToModelElements

	Column Name	Data Type	Length	Allow Nulls
🔑	pkey	int	4	
	ID	uniqueidentifier	16	
	DiagramID	uniqueidentifier	16	
	ModelElementID	uniqueidentifier	16	
	PropertyExpression	varchar	50	

Figure 7. DiagramsToModelElements Table

Once a relation is created between two model elements *ID* of the relation, *DiagramID*, *RelationTypeID*, Source and Target element's *ID*s are recorded in the Relations Table. *DiagramID* matches with the *DiagramID* in DiagramsToModelElements Table. *SourceModelElementID* and *TargetModelElementID* matches with the *ID* in DiagramsToModelElements Table.

Relations

Column Name	Data Type	Length	Allow Nulls
pkey	int	4	
ID	uniqueidentifier	16	
DiagramID	uniqueidentifier	16	
RelationTypeID	uniqueidentifier	16	
SourceModelElementII	uniqueidentifier	16	
TargetModelElementII	uniqueidentifier	16	
PropertyExpression	varchar	4000	
Name	varchar	255	
Description	varchar	4000	✓
SourceMultiplicity	varchar	50	✓
TargetMultiplicity	varchar	50	✓
CreationDate	datetime	8	
UpdateDate	timestamp	8	
ApproveDate	datetime	8	✓
PreApprovalDate	datetime	8	✓
Version	varchar	50	
CreatorID	uniqueidentifier	16	
PrivacyLevelID	uniqueidentifier	16	

Figure 8. Relations Table

Database Diagram is given in Figure 9.

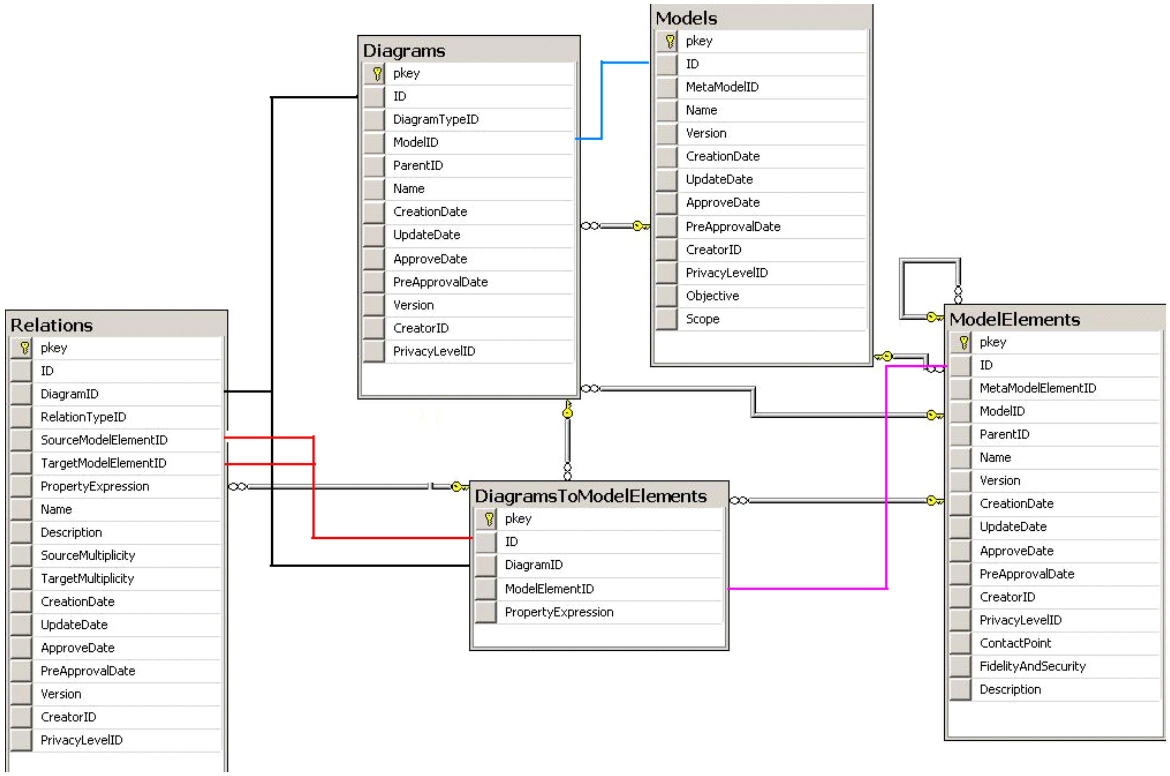


Figure 9. Database Diagram

3.2 The Automated Quality Measurement of Business Processes in the COSMOS Tool

3.2.1 Overview

In this part the Automated Quality Measurement (AQM) tool is presented. In the following sections of this part, the algorithms are explained for implementing the AQM extension for COSMOS tool. These algorithms are created within the approach of automating the measures. The algorithms are using the meta-model element's relations for retrieving the necessary data. The codes of the algorithms are given in Appendix A.

A new tab "Quality" in the menu bar is added as shown in Figure 10. Clicking on the Quality tab brings new windows form which the user selects the model as shown in Figure 11. After selecting the model, measurement results for the selected model comes in new windows form as shown in Figure 12.

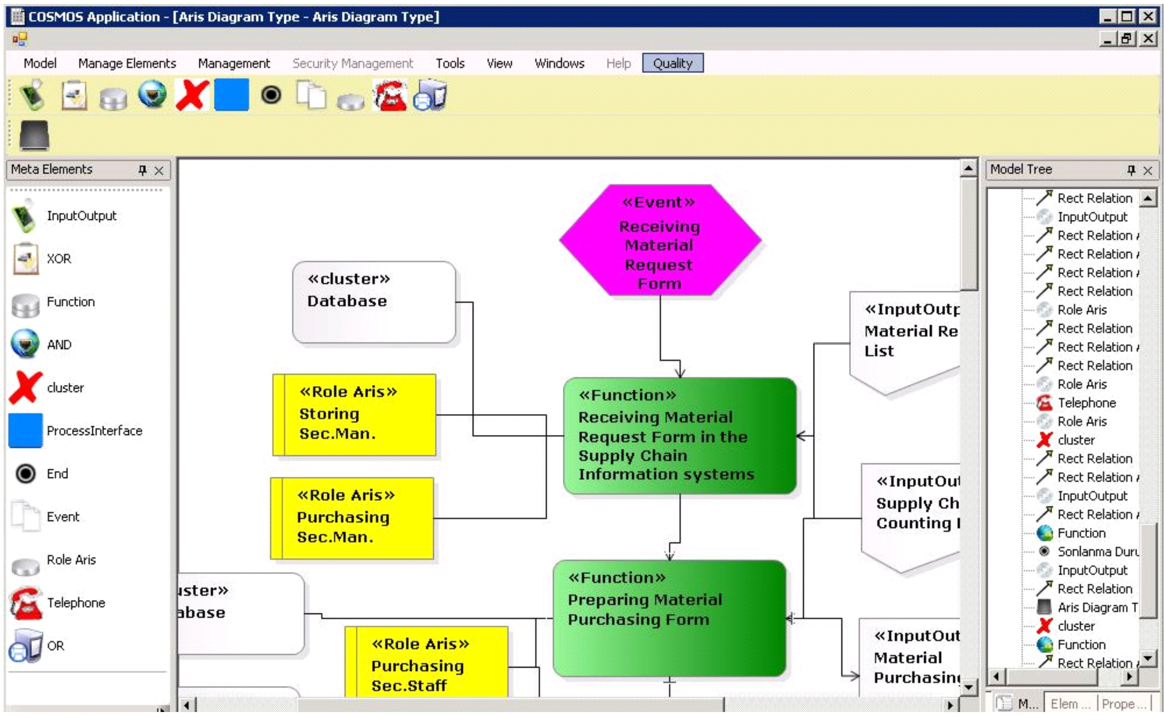


Figure 10. Extended COSMOS Tool

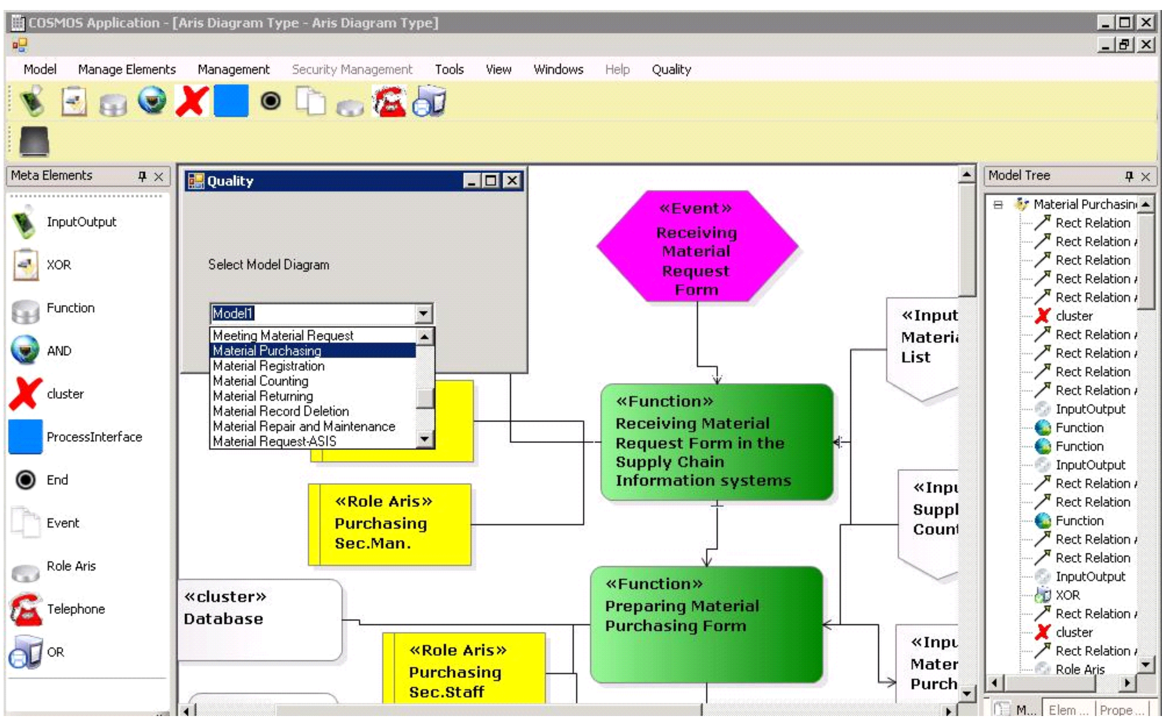


Figure 11. Selecting Model from Quality Tab

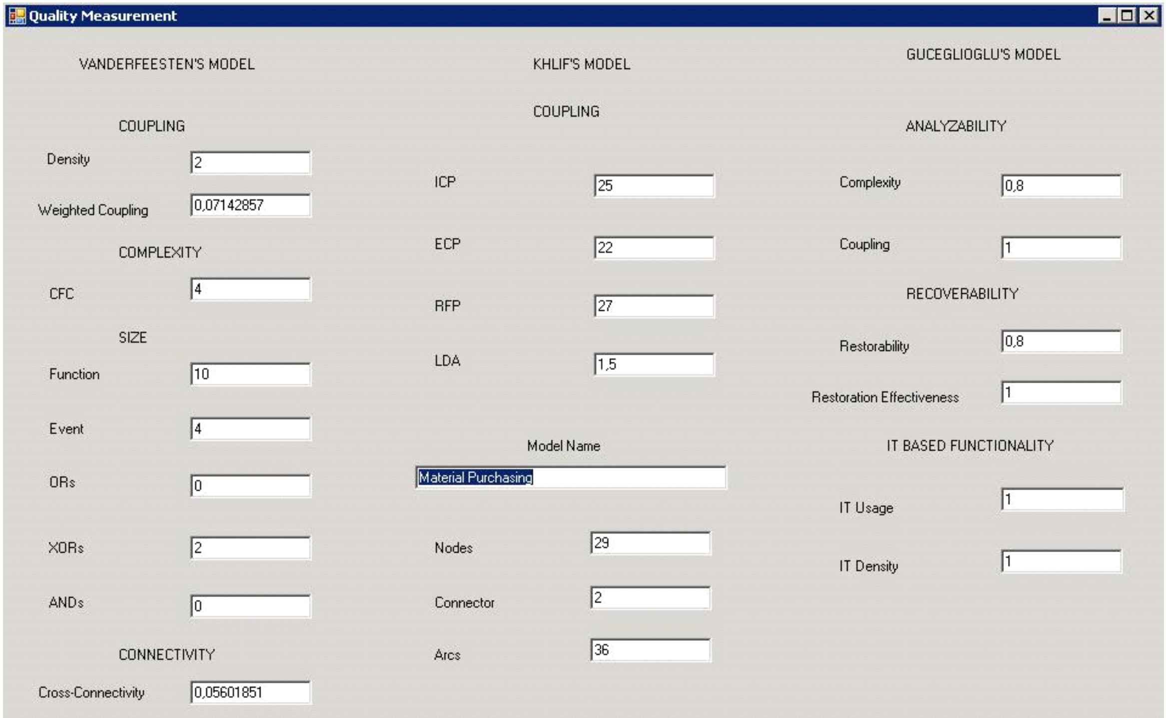


Figure 12. Measurement Results

3.2.2 Vanderfeesten's Measures

3.2.2.1 Density

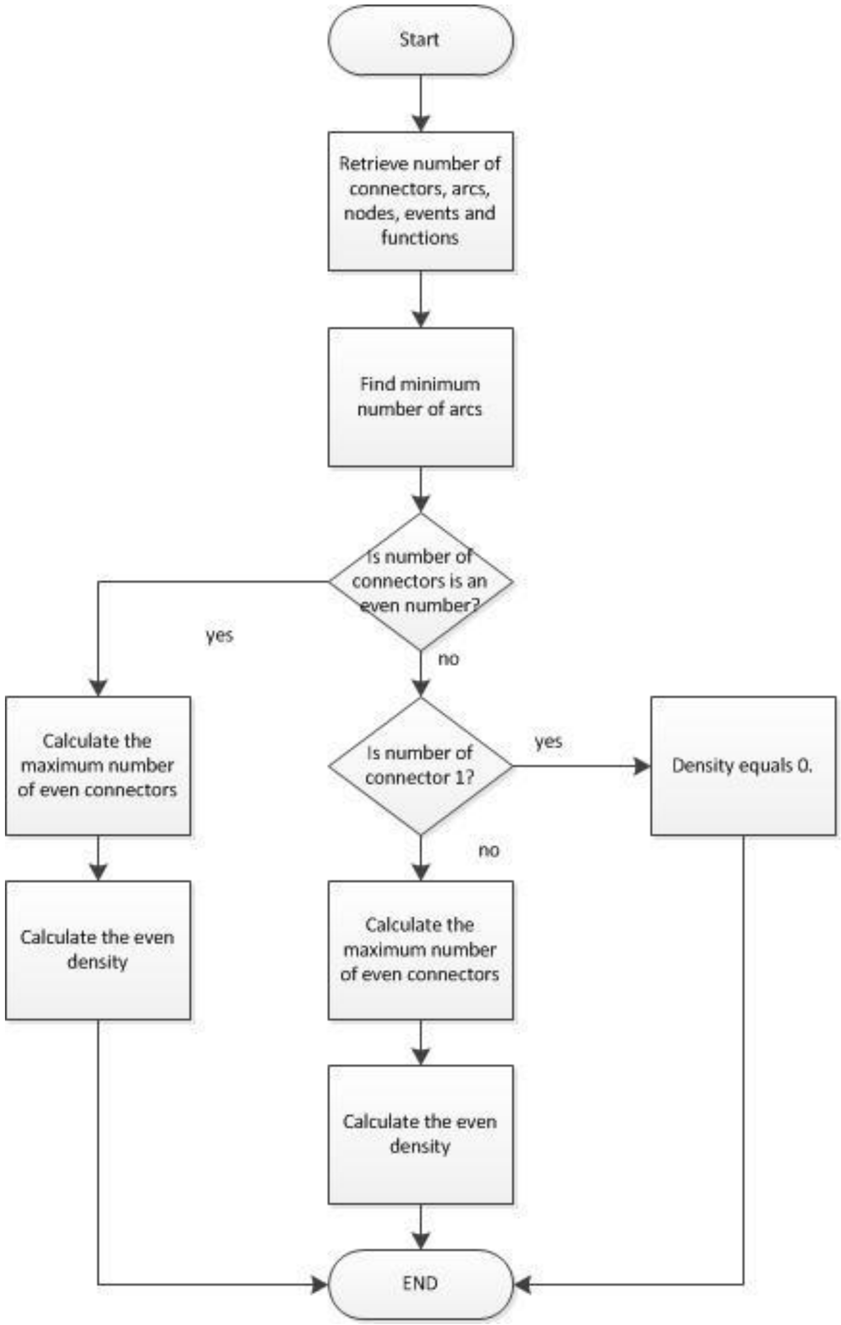


Figure 13. Denstiy Flowchart

For density calculation the needed data, such as: number of connectors, arcs, nodes, events and functions are retrieved from the database by SQL commands; *connector*, *arcs*, *nodes*, *events* and *function.* Minimum number of arcs stated in the formula is calculated by subtracting number of nodes by 1. If the number of connectors is an even number, then the maximum number of even connectors is calculated using the formula, and used in the given density formula. If the number of connectors is an odd number, then the maximum number of odd connectors is calculated using the formula, and used in the given density formula. If the number of the connector is 1 then the density is stated to be 0.

3.2.2.2 Weighted Coupling

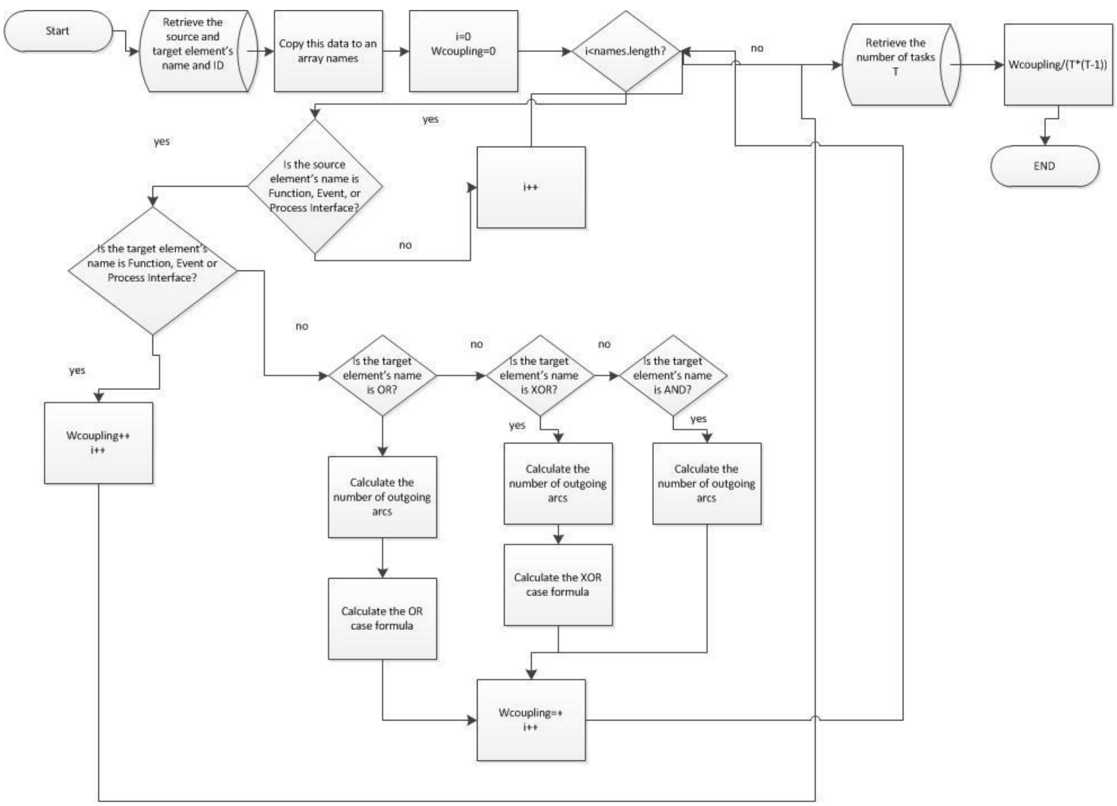


Figure 14. Weighted Coupling Flowchart

The SQL command *weighted* lists the source and target elements' name and ID. This retrieved data is then passed to the string array *names*. For each row in the array *names*, if the source element's name is Function, Event or Process Interface, target element is checked. Remind the weighted coupling formula from the Chapter 2, if the target element is again Function, Event or a Process interface, the *wcoupling* counter is increased by 1. If the target element is OR or XOR, the number of outgoing and ingoing arcs are calculated and then the formula for the OR case or for the XOR case is calculated and added to the *wcoupling* counter. Lastly if the target element is AND, then the number of outgoing arcs is found and added to the *wcoupling* counter. The number of tasks is retrieved by the SQL command *T* and used for dividing the *wcoupling* for completion of the formula.

3.2.2.3 Cross-Connectivity

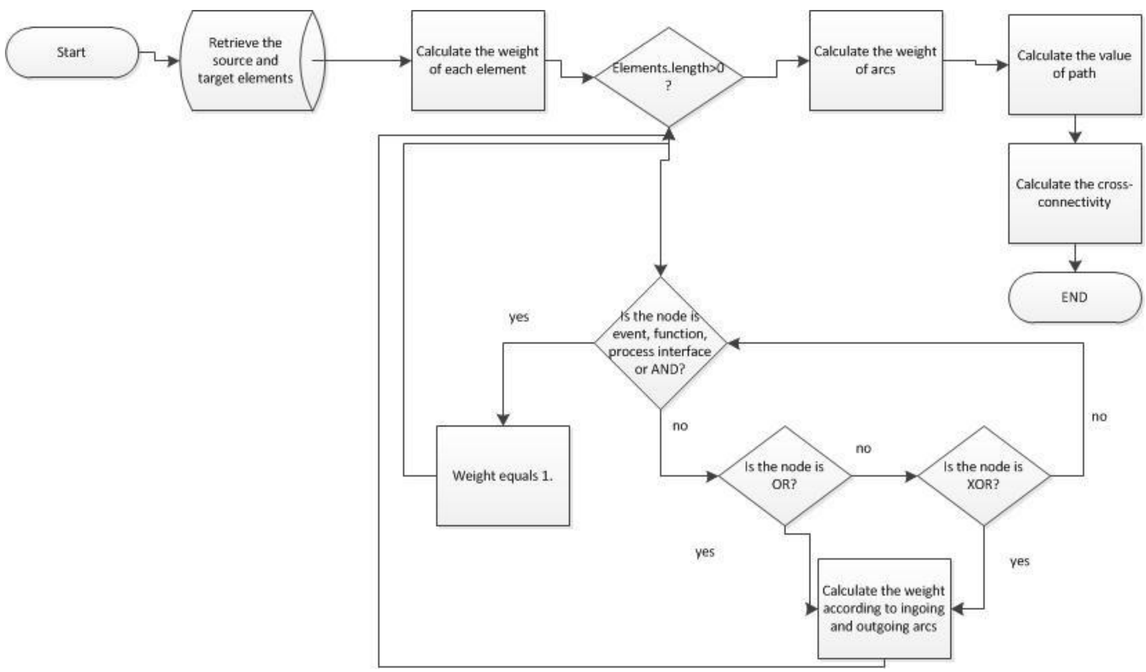


Figure 15. Cross-Connectivity Flowchart

First of all, from *Relations* table, Source and Target elements are retrieved. This is then kept in a two dimensional string array. According to the model elements, weights of the nodes are calculated. The events, functions, process interfaces and AND connector are weighted as 1 according to the formula. OR and XOR's weight depended on their ingoing and outgoing arcs. After calculating the weight of the nodes are recorded in to the two-dimensional array in the same order with the data retrieved from the Relations table. Since each of these data also represented the connecting arcs, the weight of the arc is calculated by multiplying weight of nodes which are in the same row in the *weightOfNode* array. The value of path is then calculated by adding all the values in *wightOfArc* array. Afterwards for the functions, process interfaces and events connected to each other by a connector's value is calculated and added to the value counter. At the end, for calculating the cross-connectivity, this value counter is divided by the multiplication of the total number of task and its one minus.

3.2.2.4 Control-Flow Complexity

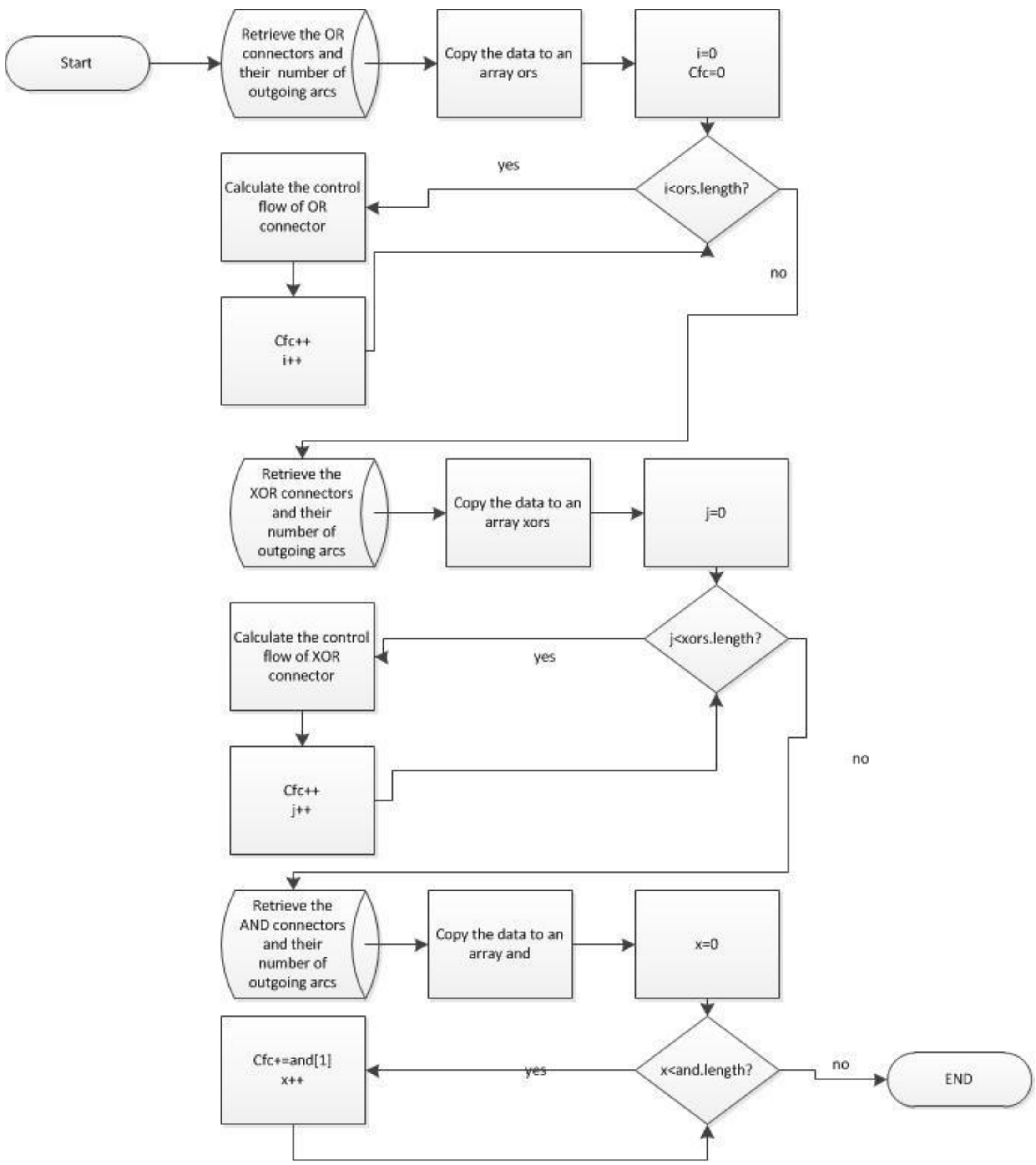


Figure 16. CFC Flowchart

For each OR connector, the number of outgoing arcs is found by the SQL command *OrOut*. This command counts the number of each OR that is in the source element place in the Relations Table. While reading the counted number for each OR, this number is used for finding the CFC of OR connector by using the formula and added to the *cfc* counter.

Also for each XOR connector the number of outgoing arcs is found by the SQL command *XorOut*. This command counts the number of each XOR that is in the source element place in the Relations Table. While reading the counted number for each XOR, this number is added to the *cfc* counter.

Lastly for AND connectors, the total number of AND connector's outgoing is found by the *AndOut* SQL command. This command counts the number of each AND that is in the source element place in the Relations Table. This number is then added to the *cfc* counter.

3.2.2.5 Size

The number of functions, events, ANDs, ORs and XORs are retrieved with the SQL commands *events, function, ANDs, ORs* and *XORs*.

3.2.3 Khlif's Measures

3.2.3.1 Imported Coupling of a Process

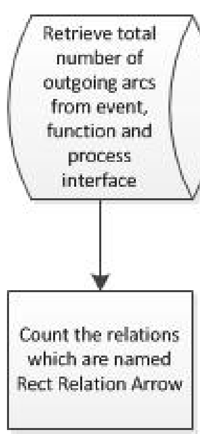


Figure 17. ICP Flowchart

Remind from the definition for the ICP measure, the total number of outgoing arcs from event, function and process interface makes up the ICP number. From the Relations Table, the count of the source elements which are event, function and process interface and whose relation name is "Rect Relation Arrow" is selected.

3.2.3.2 Exported Coupling of a Process

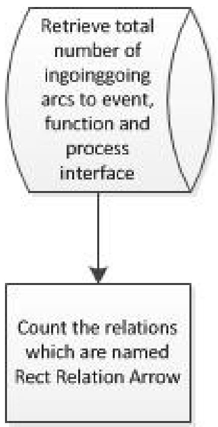


Figure 18. ECP Flowchart

Unlike the ICP measure, the total number of ingoing arcs to the event, function and process interface makes up the ECP number. From the Relations Table, the count of the target elements which are event, function and process interface and whose relation name is "Rect Relation Arrow" is selected.

3.2.3.3 Response for a Process Coupling

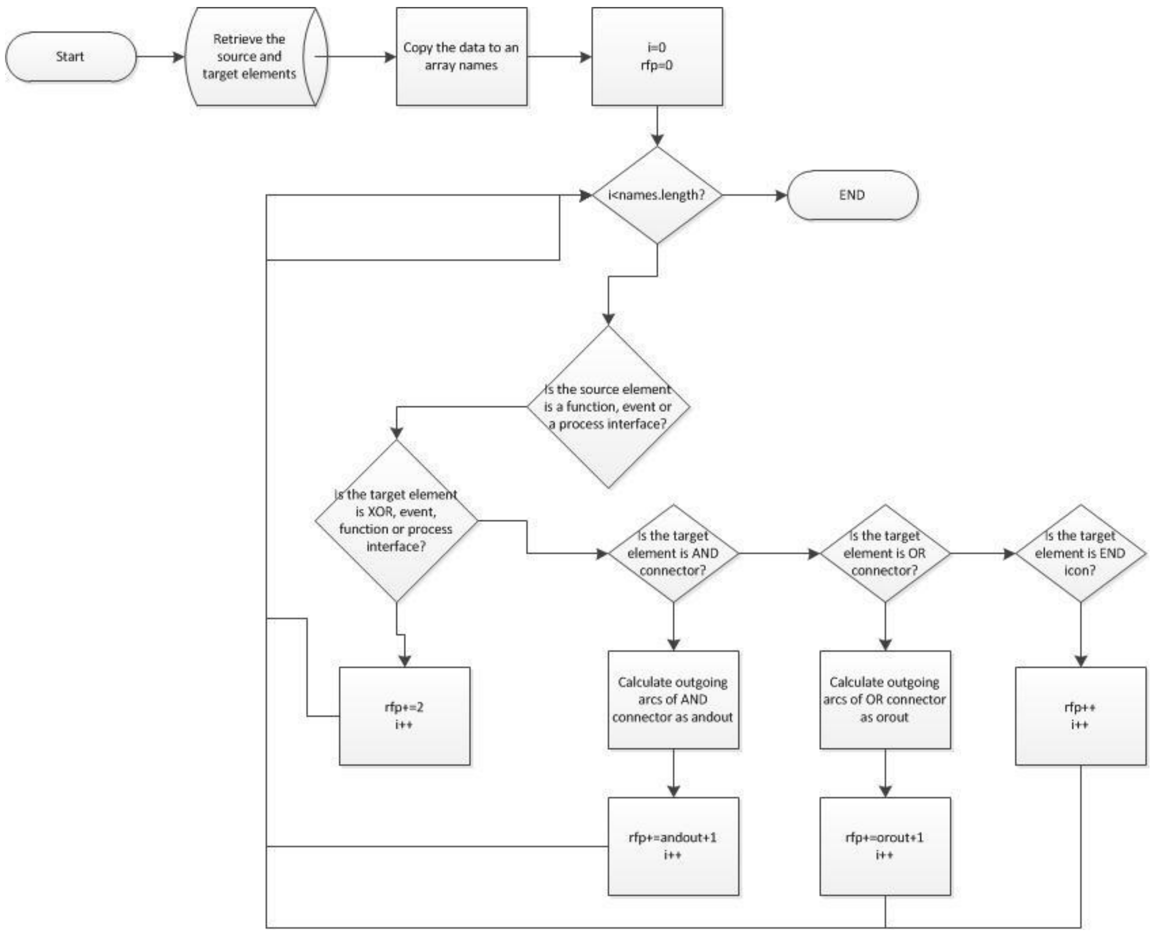


Figure 19. RFP Flowchart

For the RFP measure, remind the array named *names* from Weighted Coupling. This array is holding relation between the model elements. The first column is the name of the source element, the second column is the id of the source element, the third column is the name of the target element and lastly the fourth column is the id of the target element.

In the calculation of RFP measure, there is a counter for recording the RFP values named as *rfp*. For each row in the names array, the source element name is checked. If the source element is a function, an event or a process interface then the target element is checked. If the target element is a XOR connector, a function or an event, then the *rfp* value is increased by 2. Remind from the formula, RFP gives a value 1 to a task, and another value 1 to the next task that it has invoked. Different from the XOR connector, if the target element is a OR or an AND connector, then with the *count* method, number of outgoing arcs of the connector is found. Then the value plus 1 is added to the *rfp* counter. Remind from Chapter 2, this difference comes from the behaviors of the connectors. From the XOR connector only 1 task out of several tasks can be chosen. This is why when a task is connected to a XOR task, it can only invoke one task, and with its own value it can add only 2 to the *rfp* counter. From the OR and AND connectors one or more tasks can be processed. Thus, when a task is connected to the OR or to the AND connector it can invoke all of the tasks that are connected to the related connector. This is why the number outgoing arcs are calculated to find out how many tasks it is connected to. This number is then added to the *rfp* counter with a plus 1 value for the task that invoked them. Lastly if the task is connected to the END icon it only gets value for itself, so *rfp* value is increased by 1.

3.2.3.4 Locality of a Data Coupling

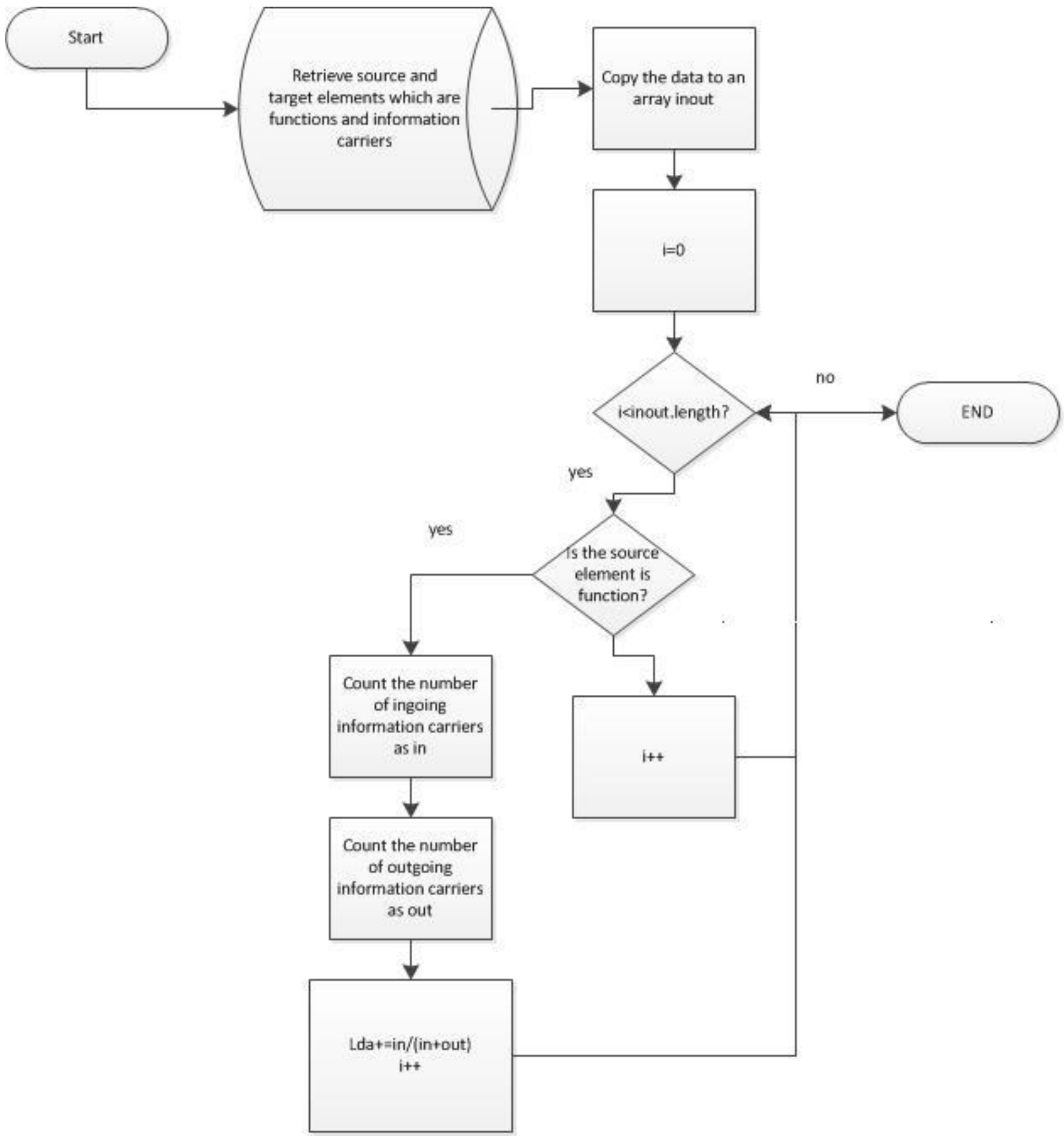


Figure 20. LDA Flowchart

The SQL command *lda* retrieves the union of source and target elements which has ingoing and outgoing information carriers. This data is then passed to the *inout* array. For each row in the array, the source element is checked if it is a function. Because the rows in the array are not distinct, with *iscounted* method, it is checked that if the element is considered before. If the element is a function then with the *count* method, the number of times that this function is seen in the source element column (which gives the number of outgoing information carrier from the function) and the number of times that this function is seen in the target element column (which gives the number of ingoing information carrier to the function) is found. Then with the formula LDA is calculated. This element is then recorded into *temp* array for not to be considered more than once.

3.2.4 Guceglioglu's Measures

3.2.4.1 Complexity

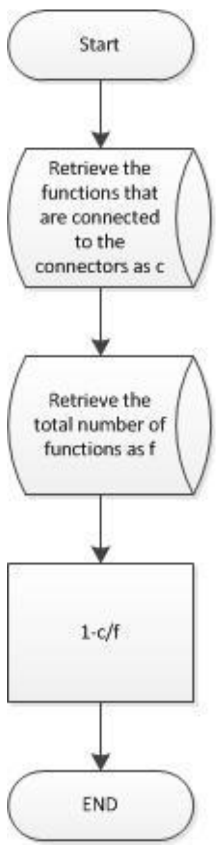


Figure 21. Complexity Flowchart

For the Complexity measure from the Relations Table, the functions that are connected to the connectors (OR, AND, XOR) is counted with the SQL command *complexity*. The SQL command *totalactivity* is for counting total numbers of functions and process interfaces in the model. This number will also be used in the following parts. Remind from the formula, complexity number is divided by the total activity number and subtracted from 1.

3.2.4.2 Coupling

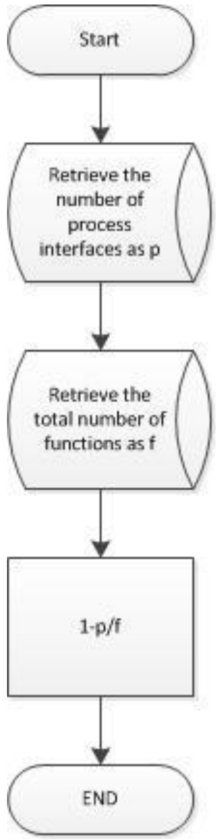


Figure 22. Coupling Flowchart

For the Coupling measure, the number of process interfaces in the model is counted by the SQL command coupling. This number is than divided by the total number of activities and subtracted by 1.

3.2.4.3 Restorability

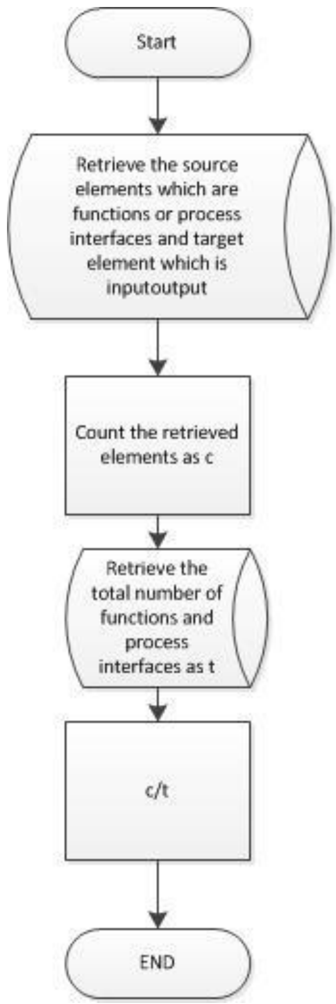


Figure 23. Restorability Flowchart

The SQL command *restorability* retrieves the count of source elements which are functions or process interfaces and whose target element is InputOutput. This number is then divided by the total number of activities.

3.2.4.4 Restoration Effectiveness

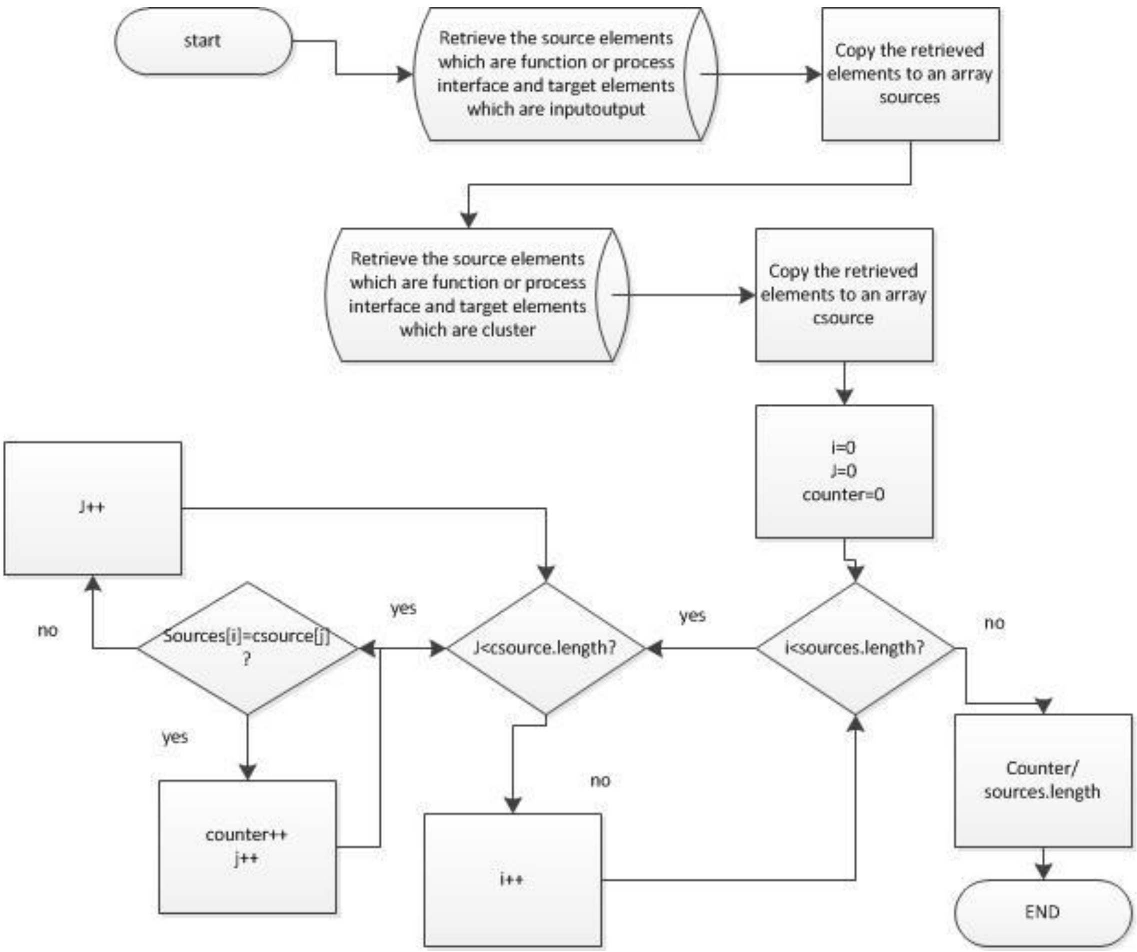


Figure 24. Restoration Effectiveness Flowchart

The SQL command *restoration* selects the source elements which is function or process interface and whose target elements are InputOutput. This data is then passed to the *sources* array. The SQL statement *clusterr* selects the source elements which is function or process interface and whose target elements are cluster. This data is then passed to the *csource* array. These two arrays *sources* and *csource* are then passed to the *isIn* method to find out how many of the source elements in the *sources* array is also in the *csource* elements. Remind from the formula, restoration effectiveness is calculated by the number of activities who has outgoing information carrier and has interaction with the cluster icons. This number is then divided by the *rest*, which is founded during calculating Restorability measure.

3.2.4.5 IT Usage

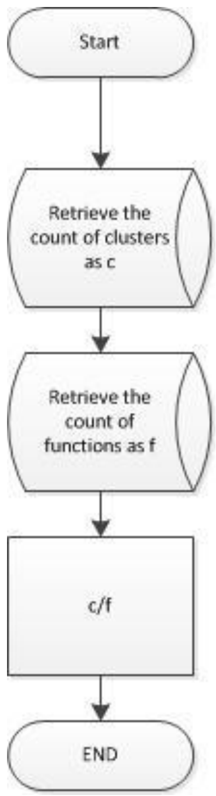


Figure 25. ITU Flowchart

This measure is calculated by finding the number of cluster icons in the model by the SQL command *ITU*. This number is then divided by the total number of activities.

3.2.4.6 IT Density

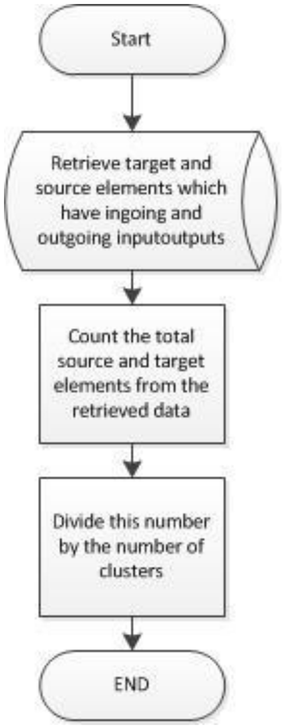


Figure 26. ITD Flowchart

The SQL command *ITD* selects the union of target and source elements which has ingoing or outgoing information carriers. This data is then passed to the *itds* array and the count of the activities which has ingoing and outgoing InputOutput is found with the *countt* counter. The *itu*, the number of clusters, which was found in the ITU measure is divided by this number and ITD is found.

CHAPTER 4

APPLICATION OF THE AUTOMATED QUALITY MEASUREMENT

This chapter describes the two case studies that were conducted as a part of this thesis. The aim of the first case study is to explore the quality measures defined for business processes in order to identify their applicability, automation potential and required time and effort. As the result of findings of the first case study, the business process quality measures which have automation potential are selected and developed an Automated Quality Measurement (AQM) tool. The second case study aims to validate the AQM tool in terms of its required time and effort to measure the quality and its accuracy.

This chapter has been organized as follows; the first section gives the research questions, the second section gives the case study designs and plans, the third section gives the implementation of the first case study, the fourth section gives the implementation of the second case study, the fifth section gives the results and discussion of the case studies and the last section gives the threads to validity.

4.1 Research Questions

Research Question 1: How can business process quality measures be automated?

The aim is to develop an integrated tool to calculate the available business process quality measures automatically. With this research question the measures that have automation potential are identified.

Research Question 2: Does the measuring quality of business processes automatically decrease the required time and effort and provide more accurate results?

The aim is to measure quality of business process more accurately with least time and effort, since manual measuring is always error prone and requires more time and effort.

Two case studies were conducted in order to answer the research questions. The first case study answers the first question. This case study explores the available measures in the literature and finds which can be automated.

The second case study answers the second question. This case study validates the automation tool in order to answer the second research question.

4.2 Case Study Design

The first case study explores how to automate the available measures from the literature. The first case study aimed to apply all these measures from different frameworks and compare the frameworks in applicability and automation potential of the measures and effort required when applying (Gurbuz 2011).

This case study was an exploration study that searched the available measures in the literature which were applicable to eEPC diagram notation, and also to observe their automation potential for conducting the second case study. According to the results of the first case study, the measures that have automation potential are then automated.

The second case study is a validation study to answer the second research question. The measures selected as an output of the first case study are automated. The Automated Quality Measurement (AQM) extension is integrated in to the COSMOS Tool which was discussed in the previous chapter. For the validation of the AQM, the aim was to answer the research question two. The measures that have automation potential are applied to the Supply Chain Management (SCM) processes manually and automatically. The results of the application and time and effort required for the application are compared for validating the automation tool.

The implementation plan for the first case study is listed as;

- Planning the literature review to identify singular measures or measures defined as part of a framework in the literature. It was planned to restrict the survey on measures that are applied to the process models that are modeled with eEPC notation.

- Selecting business process models for the application. The processes are selected according to their modeling notation and the icons used in the process models. Some of the measures need information carrier and cluster icons for calculation.

- Applying the measures on the selected processes. While each application of the measures aim was to use only the diagrams. The measures that needed further information are noted as not applicable. For the other measures some strategies are developed for the application.

- Comparing the results of the application in terms of applicability, automation potential and required time and effort. The frameworks are compared with each other by the number of measures of each framework's applicability and automation potential proportion. The required time and effort is also recorded for each application of measures.

- Analyzing the process improvement action lists prepared as part of the Regional Development Agency's Process Modeling and Analysis Project (Coskuncay 2010). The lists of processes improvement actions which are covered by the measured results are evaluated.

The implementation plan for the second case study is listed as;

- Selecting business process models for the application. The processes are selected according to their modeling notation and the icons used in the process models. Some of the measures need information carrier and cluster icons for calculation.

- Modeling the selected processes on the COSMOS tool. For the automated measurement business processes needed to be modeled in the COSMOS tool so that the data can be recorded in the COSMOS' local database. Processes are remodeled in the COSMOS environment.

- Selecting the business process quality measures which have automation potential found in the first case study.

- Applying the measures on the selected processes manually. For each application of the measure the strategy developed in the first case study is used. While the application required time is recorded.

- Developing the AQM tool and applying the measures on the selected processes manually. While the application required time is recorded.

- Comparing the results of manual and automatically measurements in terms of required effort and time and the accuracy.

The details of the case studies will be discussed in the following parts. The case study results and discussion is given at the end of this chapter.

4.3 Case Study 1: Exploration of the Measures

4.3.1 Overview

There were several measures defined in the literature which were singular and framework dimensioned. Most of the research studies were focused on singular measures defined for a specific attributes such as complexity (Cardoso 2006, Gruhn 2006), density (Mendling 2006), weighted coupling (Vanderfeesten 2007b) and cross-connectivity (Vanderfeesten 2008). For a more comprehensive approach some researchers defined quality measures in a framework dimension. These were Vanderfeesten (Vanderfeesten 2007a), Khlif (Khlif 2009) and Guceglioglu's (Guceglioglu 2006) frameworks. For the organizations that are willing to measure the quality of their business processes, it is thought that measures visualizing quality from different perspectives would be more helpful. For as, Vanderfeesten analyze quality in coupling, complexity, cohesion, modularity and size principles adapted from software programs. Khlif analyze quality from coupling and cohesion perspectives which are adopted from object-oriented programs. On the other hand Guceglioglu introduces process quality with maintainability, reliability, functionality and usability measures adapted from software quality. It is thought that all together with these frameworks, an organization can overview their business processes' quality before implementing them. A singular dimensioned released measure, cross-connectivity

(Vanderfeesten 2008), is later added to the Vanderfeesten's model for representing the cohesion perspective of the quality. The measures in the literature are given in the Table 4.

Table 4. The Available Measures

Vanderfeesten's Measures	Khlif's Measures	Guceglioglu's Measures
• Density • Weighted Coupling (WC) • Cross-Connectivity (CC) • Control-Flow Complexity (CFC) • Size (Number of Functions, Events, ORs, XORs and ANDs)	• Imported Coupling of a Process (ICP) • Exported Coupling of a Process (ECP) • Response for a Process Coupling (RFP) • Locality of a Data-Based Coupling (LDA) • Tight Process Cohesion (TPC) • Loose Process Cohesion (LPC)	• Complexity (CX) • Coupling (CP) • Failure Avoidance (FA) • Restoration (R) • Restoration Effectiveness (RE) • Functional Avoidance (FA) • Functional Completeness (FC) • IT Usage (ITU) • IT Density (ITD) • Computational Accuracy(CA) • Data Exchangeability (DE) • Access Auditability (AA) • Functional Understanding (FU) • Existence in Documents (EID) • Input Validity Check (IVC) • Undoability (U) • Attractive Interaction (AI)

4.3.2 Implementation

After the literature review of the available business process quality measures, all of the measures given in Table 4 are applied on processes of Turkish State Planning Organization's (SPO) Human Resource Management (HRM) in order to identify their applicability, automation potential and required time and effort. HRM processes were modeled in regulation of SPO for Regional Development Agency's Process Modeling and Analysis Project (Coskuncay 2011). These processes were modeled with Extended Event-Driven Process Chain (eEPC) notation. It includes 11 main processes and 10 sub-processes. The selection criteria for the processes were that it should have been modeled using eEPC diagram notation. The aim was to apply the measures on a set of processes by using only the diagram. eEPC notation is chosen for the reason that it shows information flow between activities such as documents, records and clusters. One other criterion for selecting these processes was that, it was a module based and included process improvement action list, prepared as part of the Regional Development Agency's Process Modeling and Analysis Project (Coskuncay 2010). The list of process improvement actions are evaluated in order to see if they were covered by the measured results (Gurbuz 2011). Number of process improvement actions (sub processes are not included) are given in Appendix B, according to the processes. The process properties including number of nodes, arcs, connectors and process interfaces are summarized in Table 5.

Table 5. Human Resource Management Processes

NO	Processes	Nodes	Arcs	Connector	Process Interface
1	Human Resources Planning	70	77	10	4
2	General Secretary Employment	99	105	12	0
3	Personnel Employment	83	100	14	0
4	Starting Personnel to Work	68	72	8	3
5	Permanent Charging	41	42	5	1
6	Temporary Charging	55	63	13	3
7	Attorneyship	23	25	3	1
8	Devolution of Authory	15	14	1	1
9	Giving Work Order	14	14	1	0
10	Entering Effort Record for Activity	7	6	1	0
11	In-Service Training	58	64	8	4
12	Making the In-Service Training	34	37	5	0
13	Orientation Training	19	20	2	1
14	Payed and Non-Payed Leaves	29	32	3	1
15	HR Performance Evaluation	136	161	17	1
16	Personnel Salary Calculation	5	4	0	0
17	Ending Employment Contract	50	56	11	1
18	Work Health and Security	29	34	7	2
19	Work Ethic	29	32	4	1
20	Employee Pleasure	22	24	1	0
21	Managing Announcement	10	9	0	0

After selection of the processes, each measure is applied to all of the processes of Human Resource Management. The application details of the measures are summarized in Table 6. The results of applications in terms of applicability, automation potential and required time are summarized in Table 7. Some of the measures were not applicable using only eEPC diagrams and needed further information. These measures were classified as not applicable and did not have automation potential. While application of the measures on the process model set, required time is recorded. For some of the measures, due to their complex formulas, the required time was more than the other measures. Because of the complex formulas it can be observed that the manual measurement was error-prone since it required mathematical calculations. The application is done by one person and reviewed by two people who are working in this research area.

Table 6. The Application Details

Measures	**Application**
Density	Events, functions, process interfaces, information carriers and connectors are considered as nodes. Then the connectors, arcs, functions and events are counted and integrated into the formula.
Weighted Coupling	Events, functions and process interfaces are considered as tasks for the formula.
Cross-Connectivity	Functions, events and process interfaces are considered as tasks for the formula.
Control-Flow Complexity	XORs', ORs' and ANDs' outgoing arcs are counted.
Size	Number of functions, events, ORs, XORs and ANDs are counted.
Imported Coupling of a Process	Events', functions' and process interfaces' outgoing arcs are counted.
Exported Coupling of a Process	Events', functions' and process interfaces' ingoing arcs are counted.

Table 6 (cont.)

Response for a Process Coupling	While applying this metric an activity is considered as value 1 and its connected activity as another value 1. So RFP of that activity equaled to value 2. In case where the activities were connected by a connector, value calculations varied according to the connector type. As it is indicated in section 2, if the connector is AND or OR then it is counted 1 value for each connected activity, since all of the activities are executed. If the connector is XOR, one of the activities is only counted as value 1, since only one of the activities is executed.
Locality of a Data-Based Coupling	For each function which have both ingoing and outgoing information carriers the proportion of inputs to both inputs and outputs is counted.
Tight Process Cohesion	This measure cannot be applicable since it requires at least two tasks within an activity.
Loose Process Cohesion	This measure cannot be applicable since it requires at least two tasks within an activity.
Complexity	This measure calculates the number of decision points over the total number of activities and subtracts it from 1. Applying this measure on our EPC modeled process; we count the number of connectors which are connected to functions as decision points.
Coupling	Coupling measure counts the interactions of the process with other processes over the total number of activities and subtracts it from 1. Process interfaces in a process are considered as interactions.
Failure Avoidance	Failure Avoidance calculates the number of activities in which review, inspection, checkpoints are applied over the number of activities. It is considered that events in our model are indicating checkpoints where the activity had happened or not. Such as, an event coming after activity named “Preparing a report” and indicating “Report Prepared” means it has a control point and checks if it is done and then continues to the flow.
Restorability	Restorability indicates the number of recorded activities over the number of activities. The activities whose outputs are information carriers and indicate a list, form or a document are counted.

Table 6 (cont.)

Restoration Effectiveness	Restoration Effectiveness indicates the number of activities which can be restored over the number of recorded activities are calculated. While the application, the activities whose outputs are folder and cluster icons counted as restored activities and divide the total by the number of recorded activities.
Functional Avoidance	Functional Adequacy defines if there is unconformity between the activity in the practice and the activity defined in the related documents. This measure was not applicable since it required further information than required from diagrams.
Functional Completeness	Functional Completeness identifies if there are missing activities in process according to regulatory documents. This measure was not applicable since it required further information than required from diagrams.
IT Usage	Guceglioglu defines IT Usage for to calculate the number of activities in which IT applications are used for creating, deleting, updating or searching purposes over the number of activities. Our process models have cluster icons which specify the database so the activities which have interaction with these cluster icons are counted.
IT Density	Guceglioglu defines IT Density for to calculate the number of forms, reports or other documents which are prepared, updated, deleted or searched by using IT applications over the number of forms, documents in the process. While calculating this measure the cluster icons are counted and divided by the number of total forms, documents and archival records.
Computational Accuracy	Computational Accuracy measures the implementation of the accuracy requirements in process design. This measure was not applicable since it required further information than diagrams.
Data Exchangeability	The Data Exchangeability measure specifies the operations applied to the data received from other processes. This measure was not applicable since it required further information than diagrams.

Table 6 (cont.)

Access Auditability	The Access Auditability measure is defined for auditing access to process activities so that any person who accessed to the data can be analyzed. This measure was not applicable since it required further information than diagrams.
Functional Understandability	The Functional Understandability measure is used for specifying difficulties for understanding activities. During the application of this measure, the understandability is identified by reading the activity names.
Existence in Documents	The Existence in Documents measure is used for analyzing the available documents about the model and measures which of the activities are defined in them. This measure was not applicable since it required further information than diagrams.
Input Validity Check	Input Validity Checking calculates the number of activities in which validity checking can be performed for input parameters over the number of activities. It is assumed that the names of the activities which include approval, evaluation, preparing a report and ext. checks the validity of input.
Undoability	Undoability calculates the number of activities which can be undone. The activities in which the names include ‘updating’ are considered. It is assumed that if it is possible to update then it would also be possible to undo in case of mistake.
Attractive Interaction	This measure calculates the number activities which staff doesn’t face any difficulties over the number of recorded activities. While applying this measure the activities that the staff may face difficulties are figured out from the names of the activities.

Table 7. The Application Results

Frameworks	Applicable Measures	Automatable Measures	Required Time
Vanderfeesten	-Density -Weighted Coupling -Control-Flow Complexity -Size	-Density -Weighted Coupling -Control-Flow Complexity -Size	6 hours
Khlif	-Imported Coupling of a Process (ICP) -Exported Coupling of a Process (ECP) -Response for a Process Coupling (RFP) -Locality of a Data-Based Coupling (LDA)	-Imported Coupling of a Process (ICP) -Exported Coupling of a Process (ECP) -Response for a Process Coupling (RFP) -Locality of a Data-Based Coupling (LDA)	3 hours
Guceglioglu	-Complexity (CX) -Coupling (CP) -Failure Avoidance (FA) -Restoration (R) -Restoration Effectiveness (RE) -IT Usage (ITU) -IT Density (ITD) -Input Validity Check (IVC) -Undoability (U) -Attractive Interaction (AI)	-Complexity (CX) -Coupling (CP) -Restoration (R) -Restoration Effectiveness (RE) -IT Usage (ITU) -IT Density (ITD)	4 hours

Table 8. Measures' Automation Potential Determinations

Measures	Determination of Automation Potential
Density	The needed variables for the density measure's formula are number of events, number of functions, number of connectors, number of nodes (information carriers, events, functions, process interfaces, connectors). This data can be easily retrieved from database so this measure is found to be automatable.
Weighted Coupling	In weighted coupling measure, task is considered to be function, event or process interface. Connectors' ingoing and outgoing arcs are retrieved from the database. Afterwards this result is divided by the total number of tasks.
Control-Flow Complexity	CFC measure is based on the connector's ingoing and outgoing arcs. This data is easily retrieved from database and is able to be calculated automatically.
Size	Size measurement the needed data is number of nodes, arcs, connectors, functions and events. This data is easily retrieved from the database.
Imported Coupling of a Process and Exported Coupling of a Process	ICP measure requires the total number of outgoing arcs from the activities. ECP measure requires the total number of ingoing arcs to activities. These are kept in database so both of these measures are automatable.
Response for a Process Coupling	RFP measure needs each event, functions and process interface's connected event, function, and process interface. Retrieving this data from database and making the calculation can be automated.
Locality of a Data-Based Coupling	LDA measure needs each function's or process interfaces' ingoing and outgoing information carriers. This data can be retrieved and automated for the calculation.
Complexity	Complexity measure is calculated by the decision points in which eEPC diagrams it is notated as the connectors. Connectors that are connected to the functions are recorded in the database. Retrieving this data will make the measurement automated.

Table 8 (cont.)

Coupling	Coupling measure is based on the process interfaces. This measure is also automatically calculated by retrieving the number of process interfaces.
Restorability	Restorability measure is calculated by finding the number of functions or process interfaces which has outgoing information carriers. Since this data is kept in the database, restorability measure can be calculated automatically.
Restoration Effectiveness	In this measure, it is thought that if the information carriers are recorded in a computer environment then they can be effectively restored. For calculating this measure, the number of functions or process interfaces which have outgoing information carriers and have interaction with a database icon is found and it is divided by the total number of functions or process interfaces which have outgoing information carriers. Since both of these data is kept in the database, this measure can be calculated automatically.
IT Usage	For this measure the functions or process interfaces which have interaction with a database icon is needed. Since this data is kept in the database, this measure can be calculated automatically.
IT Density	For this measure, the number of database icon and number of functions or process interfaces which have ingoing or outgoing information carriers are needed. Since both of these data is kept in the database, this measure can be calculated automatically.

4.3.3 Results

In the first case study, the aim is to identify the measures which have automation potential. 3 frameworks and one singular measure have been chosen from the literature. The case study has been done by one person. 4 of Vanderfeesten's 5 measures are applicable on eEPC diagrams and have potential of automation. While applying these measures no difficulties were faced. The reason that the other metric (cohesion) could not applied is that it is defined for workflow notation and not for eEPC. 4 of Khlif's 6 measures are applicable on eEPC diagrams and have potential of automation. The other two measures (Tight Process Cohesion and Loose Process Cohesion) do not fit in to the eEPC diagram notation. In the eEPC diagram each function represents one task, but TPC and LPC measures are based on a function with two tasks.

Guceglioglu's 10 of 17 measures are applicable. Other measures need high level details for application and using only eEPC diagrams is not achievable. Such as for Functional Avoidance measure which is for identifying whether or not there is any checkpoint or review in the activity. This measure is only applicable by a subjective decision by reading the name of the activity, so it cannot be automated. Another similar situation happens in Functional Understandability, Undoability, and Attractive Interaction measure. These measures need human interpretation, therefore they cannot be automated. Measures such as Functional Adequacy, Functional Completeness, Computational Accuracy, Data Exchangeability, Access Auditability, Existence in Documents and Input Validity Checking need high level documents for calculation and cannot be observed from the diagrams. As a result, from these 10 measures only 6 of them have automation potential. The remaining 4 measures need human subjective for calculation. Vanderfeesten's other measure released for cohesion (cross-connectivity) is also considered and added to the measures list. The application is done by one person. The required time for applying the measures on the Human Resources Management processes is 13 hour/man. There are many other business process model sets in Turkish State Planning Organization and therefore this quality measurement calculation will not just end by spending only 13 hours. On the other hand, business processes are continuously being improved so the quality calculations are done over and over again. Therefore, it is observed that manual calculations will continuously require time and effort.

With all these measures, different perspectives provide greater opportunities for organizations to improve their processes. For the first case study, one of our goals was to discuss the process improvement action list with the measures, and observe if there is a correlation. Measures, such as R, FA, and CP were specified in the improvement action list for processes. Even though they seemed to be improved, it was observed that they are not at the higher levels and still need to be improved. Since some of the measures (FAD, AA, CA, FC) could not be applied in our model, if they are covered in improvements cannot be concluded. Our other goal was to analyze if all these frameworks are comprehensive to measure the quality of a process. The process improvement list does not mention about coupling, cohesion and complexity of the model. These measures will be more meaningful when an improvement is made after modeling. It is observed that more comprehensive measures are needed. An example from the improvement list is that, Training and Charging is specified in the regulatory documents but it is not clear enough for the personnel to find out which process to take into consideration so that process needs to be divided into sub-processes. A measure in this situation can be helpful in deeply specifying and improving the processes. The results show that measures with different point of views enables us to observe quality from different perspectives.

4.4 Case Study 2: Validation of the Automation Tool

4.4.1 Overview

The results of the first case study showed that the application of the measures take time and require effort. Some of the measures have an automation potential; therefore it is thought that time and effort can be decreased by automating these measures with an AQM tool. After the implementation of this tool, the second case study was designed for the aim of the validation.

For conducting the second case study Supply Chain Management processes were chosen. The selection criteria was that it should be modeled using an eEPC diagram notation that included information flow between activities, documents, records and clusters. The selected process had AS-IS and TO-BE form. These two forms provide diversity to the application. The measurement results of these two forms were also thought to be helpful in highlighting the process improvement's importance by a comparison. Process activity details are given in Table 9.

Table 9. Supply Chain Management Process Activity Details

Process Name	AS-IS Form Activity Number	TO-BE Form Activity Number
Material Request	16	6
Meeting Material Request	18	6
Material Purchasing	12	10
Material Registration	5	5
Material Counting	20	12
Material Returning	14	7
Material Record Deletion	15	10
Material Repair and Maintenance	6	5

4.4.2 Implementation

After selecting the business process set, these processes were re-modeled in the COSMOS Tool. The COSMOS Tool keeps every record of activity in its database. As explained in the previous chapter, the diagrams are modeled in the tool, functions, events, information carriers and their relationship with other model elements are recorded in the database. One difficulty was finding a correct match for the model elements. For as, for the telephone icon we had to create a new model element in the diagram. Other than telephone icon, cluster icon is also added to the model elements for the diagram. The modeled diagrams in the COSMOS Tool are given in the Appendix C.

The set of processes were firstly calculated manually by one person. The calculations were based on their formulas which were described in the second chapter and also given in the Table 8. For each measure, the needed variables for the calculations are given in the Table 10.

Table 10. The measures' needed data for calculation

Measure	Data
Density	-Number of connectors -Number of arcs -Number of nodes -Number of functions and events
Weighted coupling	-Relations between the model elements
Control-Flow Complexity	-Number of outgoing arcs of OR connector - Number of outgoing arcs of XOR connector - Number of outgoing arcs of AND connector
Cross-Connectivity	-Weight of node -Weight of arc -Value of path -Value of a connection
Size	-Number of function, event -Number of XOR, OR, AND
Imported Coupling of a Process	-Number of outgoing arcs of each element
Exported Coupling of a Process	-Number of ingoing arcs of each element
Response for a Process Coupling	-Each element's invoked element
Locality of a Data-Based Coupling	-For each element, number of incoming input over number of outgoing input and incoming input
Complexity	-Number of connectors that are connected to the functions -Number of activities

Table 10 (cont.)

Coupling	-Number of process interfaces -Number of activities
Restoration	-Number of activities that have outgoing outputs
Restoration Effectiveness	-Number of clusters - Number of activities that have outgoing outputs
IT Usage	-Number of clusters -Number of activities
IT Density	-Number of clusters -Number of activities which have ingoing and outgoing input-outputs

The manual calculations of the measures are given in the Tables 11, 12, 13, 14, 15, 16.

Table 11. Vanderfeesten Measures on ASIS Processes

Processes	**Nodes**	**Arcs**	**Function**	**Events**	**Connecto**	**Density**	**Weighted Coupling**	**CFC**	**CC**
Material Request	**37**	37	16	3	1	0,000	0,497	2	0.045
Meeting Material Request	**45**	48	18	3	2	2,000	**0,050**	4	0.038
Material Purchasing	**32**	**37**	12	3	1	0,000	0,067	2	0.059
Material Registration	12	**14**	4	1	1	0,000	0,133	2	0.100
Material Counting	**46**	**52**	20	0	3	Infinity	**0,050**	6	0.038
Material Returning	30	**34**	14	0	1	0,000	**0,060**	2	0.058
Material Record Deletion	**32**	39	15	0	2	**2,667**	0,057	4	0.045
Material Repair and Maintenance	13	**15**	6	1	1	0,000	**0,119**	2	0.111

Table 12. Khlif's Measures on ASIS Processes

Processes	**ECP**	**ICP**	**LDA**	**RFP**
Material Request	**24**	30	0,500	36
Meeting Material Request	**28**	35	1,000	40
Material Purchasing	**22**	**26**	1,500	29
Material Registration	11	**5**	0,000	10

Table 12 (cont.)

Material Counting	**25**	38	**3,333**	39
Material Returning	**19**	**25**	2,333	26
Material Record Deletion	**23**	**21**	2,833	27
Material Repair and Maintenance	9	**11**	0,500	14

Table 13. Guceglioglu's Measures on ASIS Processes

Processes	Complexity	Coupling	Restorability	Restoration Effectiveness	ITU	ITD
Material Request	0,938	1	0,688	0,545	0,375	0,462
Meeting Material Request	0,889	1	**0,500**	**0,333**	0,278	**0,357**
Material Purchasing	0,917	0,800	**0,583**	**0,714**	0,417	**0,556**
Material Registration	0,800	1	0,400	0,000	0,200	0,200
Material Counting	0,850	1	0,800	0,250	0,250	0,294
Material Returning	0,929	1	**0,643**	**0,333**	**0,167**	**0,273**
Material Record Deletion	0,867	1	**0,600**	**0,444**	0,267	0,333
Material Repair and Maintenance	0,833	1	**0,667**	**0,250**	0,167	0,167

Table 14. Vanderfeesten Measures on TOBE Processes

Processes	Nodes	Arcs	Functions	Events	Connectors	Density	Weighted Coupling	CFC	CC
Material Request	17	**18**	6	3	1	0,000	**0,083**	2	0.080
Meeting Material Request	17	18	6	3	1	0,000	0,097	2	0.080
Material Purchasing	**29**	**35**	10	4	2	**1,750**	**0,146**	4	**0.051**

Table 14 (cont.)

Material Registration	16	**18**	5	3	1	0,000	0,107	2	0.086
Material Counting	**28**	**33**	12	0	3	**2,000**	**0,083**	6	0.055
Material Returning	13	18	7	0	1	0,000	0,143	2	0.111
Material Record Deletion	22	25	10	0	2	1,333	**0,067**	4	0.056
Material Repair and Maintenance	10	14	5	1	1	0,000	0,167	2	**0.100**

Table 15. Khlif's Measures on TOBE Processes

Processes	ECP	ICP	LDA	RFP
Material Request	**14**	**9**	0,000	16
Meeting Material Request	**12**	**9**	0,500	16
Material Purchasing	**21**	**19**	**1,000**	27
Material Registration	**15**	**7**	0,000	14
Material Counting	28	16	1,167	26
Material Returning	**12**	10	1,167	13
Material Record Deletion	**12**	16	**1,500**	17
Material Repair and Maintenance	10	**7**	0,000	11

Table 16. Guceglioglu's Measures on TOBE Processes

Processes	Complexity	Coupling	Restorability	Restoration Effectiveness	ITU	ITD
Material Request	0,833	1	0,167	1,000	0,833	1,000
Meeting Material Request	0,833	1	0,500	0,667	1,000	1,000
Material Purchasing	0,800	1	**0,500**	1,000	1,000	1,000
Material Registration	0,800	1	0,400	1,000	1,000	1,000
Material Counting	0,750	1	0,417	1,000	1,000	1,000
Material Returning	0,857	1	0,571	**0,750**	0,714	**1,000**
Material Record Deletion	0,800	1	0,700	1,000	0,700	0,875
Material Repair and Maintenance	0,800	1	0,400	0,500	0,800	0,800

After the manual calculation the next step was to calculate these measures automatically with the tool. The detailed information about how the tool works were explained in Chapter 3. Results of the calculations are given in the Table 17, 18, 19, 20, 21, 22.

Table 17. Vanderfeesten Measures on ASIS Processes

Processes	Nodes	Arcs	Functions	Events	Connectors	Density	Weighted Coupling	CFC	CC
Material Request	38	37	16	3	1	0,000	0,049	2	0.045
Meeting Material Request	46	48	18	3	2	2,000	0,045	4	0.038
Material Purchasing	34	39	12	3	1	0,000	0,067	2	0.059
Material Registration	12	15	4	1	1	0,000	0,133	2	0.100
Material Counting	47	54	20	0	3	Infinity	0,048	6	0.038
Material Returning	30	35	14	0	1	0,000	0,065	2	0.058
Material Record Deletion	33	39	15	0	2	3,500	0,057	4	0.045
Material Repair and Maintenance	13	16	6	1	1	0,000	0,142	2	0.111

Table 18. Khlif's Measures on ASIS Processes

Processes	ECP	ICP	LDA	RFP
Material Request	23	30	0,500	36
Meeting Material Request	30	35	1,000	40
Material Purchasing	24	28	1,500	29
Material Registration	11	6	0,000	10
Material Counting	32	38	3,833	39
Material Returning	22	24	2,333	26
Material Record Deletion	24	25	2,833	27
Material Repair and Maintenance	9	12	0,500	14

Table 19. Guceglioglu's Measures on ASIS Processes

Processes	Complexity	Coupling	Restorability	Restoration Effectiveness	ITU	ITD
Material Request	0,938	1	0,688	0,545	0,375	0,462
Meeting Material Request	0,889	1	0,555	0,400	0,278	0,333
Material Purchasing	0,917	0,800	0,667	0,625	0,417	0,454
Material Registration	0,800	1	0,400	0,000	0,200	0,200
Material Counting	0,850	1	0,800	0,250	0,250	0,294
Material Returning	0,929	1	0,571	0,375	0,214	0,272
Material Record Deletion	0,867	1	0,666	0,400	0,267	0,333
Material Repair and Maintenance	0,833	1	0,833	0,200	0,167	0,167

Table 20. Vanderfeesten Measures on TOBE Processes

Processes	Nodes	Arcs	Function	Events	Connecto	Density	Weighted Coupling	CFC	CC
Material Request	17	17	6	3	1	0,000	0,097	2	0.080
Meeting Material Request	17	18	6	3	1	0,000	0,097	2	0.080
Material Purchasing	29	36	10	4	2	2,000	0,071	4	0.056
Material Registration	16	17	5	3	1	0,000	0,107	2	0.086
Material Counting	27	36	12	0	3	2,500	0,079	6	0.055
Material Returning	13	18	7	0	1	0,000	0,143	2	0.111
Material Record Deletion	22	25	10	0	2	1,333	0,077	4	0.056
Material Repair and Maintenance	10	14	5	1	1	0,000	0,167	2	0.124

Table 21. Khlif's Measures on TOBE Processes

Processes	ECP	ICP	LDA	RFP
Material Request	13	10	0,000	16
Meeting Material Request	13	11	0,500	16
Material Purchasing	22	25	1,500	27
Material Registration	14	8	0,000	14
Material Counting	28	16	1,167	26
Material Returning	13	10	1,167	13
Material Record Deletion	14	16	1,667	17
Material Repair and Maintenance	10	8	0,000	11

Table 22. Guceglioglu's Measures on TOBE Processes

Processes	Complexity	Coupling	Restorability	Restoration Effectiveness	ITU	ITD
Material Request	0,833	1	0,167	1,000	0,833	1,000
Meeting Material Request	0,833	1	0,500	0,667	1,000	1,000
Material Purchasing	0,800	1	0,800	1,000	1,000	1,000
Material Registration	0,800	1	0,400	1,000	1,000	1,000
Material Counting	0,750	1	0,417	1,000	1,000	1,000
Material Returning	0,857	1	0,571	0,500	0,714	0,714
Material Record Deletion	0,800	1	0,700	1,000	0,700	0,875
Material Repair and Maintenance	0,800	1	0,400	0,500	0,800	0,800

The aim was to validate the tool in terms of its time, effort and accuracy. For this validation the set of processes has been chosen in AS-IS and TO-BE form. The selected measures from the first study are applied on the processes both manually and automatically. This way it was possible to observe the time and effort required for the manual and automated calculation. Most of the errors made while applying the measures on the process model set was caused by counting the nodes and arcs wrong, due to the processes' complex design. With the AQM tool, these wrong calculations are eliminated since it directly retrieves the nodes and arcs from the database. Therefore, by comparing the automatically and manual calculations accuracy is validated.

4.4.3 Results

The second research question was for validating that with an AQM, business process quality would be measured more effectively with less required time and effort and more accuracy. To answer this question, a set of processes which were modeled in an eEPC diagram notation had been chosen and the selected measures from the first study were applied manually and automatically. The required time is recorded while measuring. As summarized in Table 23 and Table 24 total effort spent for measuring the quality is found to be 149 minutes. By AQM this calculation only takes 1 second for each process so in total 16 seconds for the 2 sets of processes. Since this recorded time is dependent on the set of process' size, it still can be highlighted that it would take longer than the AQM. 149 minutes can be spent for once to calculate the quality of the business process. However, this is only one set of business processes, and there are more processes. Also, in quality management processes should be continually improved, therefore this calculations will not be done just once and the process quality will be calculated continuously.

Table 23. Time Spent for ASIS Process Manual Calculation

Processes	**Time**
Material Request	10 minutes
Meeting Material Request	13 minutes
Material Purchasing	8 minutes
Material Registration	7 minutes
Material Counting	16 minutes
Material Returning	10 minutes
Material Record Deletion	12 minutes
Material Repair and Maintenance	7 minutes
Total	**83 minutes**

Table 24. Time Spent for TOBE Processes Manual Calculation

Processes	**Time**
Material Request	7 minutes
Meeting Material Request	9 minutes
Material Purchasing	11 minutes
Material Registration	6 minutes
Material Counting	10 minutes
Material Returning	8 minutes
Material Record Deletion	10 minutes
Material Repair and Maintenance	6 minutes
Total	**66 minutes**

On the other hand, an accuracy problem with manual calculation is also validated. Not only in time and effort consideration, in manual calculation it is seen that mostly the number of nodes, arcs, weighted coupling, ICP, ECP, LDA, restorability, restoration effectiveness measures are calculated wrong. This is because of the process length and its complexity and therefore it is easy to mistake when counting the nodes and the arrows. Since in TO-BE form processes there were less numbers of activities and were less complex, error proportion is smaller according to the AS-IS form processes. This problem cannot be generalized since this error proportion is dependent on the person making the calculations. However, since the measures have complex formulas, it is always possible for the measurers to count the nodes and arrows wrong.

Another point in this case study that should be considered is that the process design quality of AS-IS and TO-BE processes. The number of nodes, arcs, functions is less in TO-BE form processes according to AS-IS form processes. On the other hand weighted coupling, ECP, ICP, LDA and RFP values are also lesser in TOBE processes according to ASIS processes as shown in Table 18 and Table 21. At this point, the difference between AS-IS and TO-BE form processes are clearly seen and once again, importance of the business process quality has been highlighted. The processes which have more quality according to the measures used can be specified.

4.5 Validity Threats

There are three threats to validity of the case studies. The first threat is that while selecting the business processes for the application, they were not selected for the case that they give good results. The only selection criterion was that the processes were modeled with eEPC notation and included the necessary icons for application of the measures.

The second threat is that the business process quality measures' were applied on the business processes which were from different area (Human Resource Management and Supply Chain Management). Therefore the application of the measures are tried to be generalized among different business processes.

The third threat is that the AQM tool is validated by only one case study. The resolution for this threat is left to the future study in which more case studies will be conducted for validation.

The fourth threat is that the AQM tool is developed to show that the measures which have automation potential can really be automated. Therefore, there are no external participants in this case study to evaluate the AQM's usability, robustness and user-friendliness.

CHAPTER 5

CONCLUSION AND THE FUTURE STUDY

This chapter gives the conclusion and discussion of the research in the first section. The future study is given in the second section.

5.1 Conclusion

In this thesis the aim is to automate the quality measurement of business processes with Automated Quality Measurement (AQM) tool. As business process modeling became popular in the organizations, their management gained significant importance. For this reason many researchers started to study on how to improve business processes. Because of the similarity of business processes and software programs, researches on this study mainly concentrated on adapting the software measures to the business processes. This adaptation developed both in singular and framework dimension. In order to provide the quality measurement in a more comprehensive way, this study takes into consideration three available frameworks in the literature. These frameworks also included some of the singular dimensioned measures in the literature. Thus the aim of this study was to provide organizations to measure their business processes comprehensively before executing them.

This research was conducted with two case studies. The first case study was an exploration study. There were many available measures in the literature defined for measuring business process quality and this study aimed to find the measures which are applicable on eEPC diagrams, have automation potential and their required time and effort. Vanderfeesten’s 6 measures, Khlif’s 6 measures and Guceglioglu’s 17 measures are applied on Human Resources

Management processes manually. Vanderfeesten's 5 measures are applicable on EPC diagrams and also 5 of them have automation potential and took 6 man/hours for the application.

Vanderfeesten's cohesion measure was defined for workflow notation; therefore it was not applicable on EPC notation. For this reason after this case study, Vanderfeesten's another singular measure defined for cohesion (cross-connectivity) is later add to the available business process quality measures list. Khlif's 4 measures are applicable and also 4 of them have automation potential. The two other measures defined for cohesion (Tight Process Cohesion and Loose Process Cohesion) are not applicable on EPC notation since each activity in a process had only one task. But these measures require at least two tasks in an activity. Khlif's measure's application took 3 man/hours. Guceglioglu's 10 out of 17 measures are applicable and from these 10 measures only 6 measures have automation potential. The remaining 7 measures need further information for the application such as regularity documents used while modeling the processes. Because the aim is to measure business process models using the diagrams, these measures are noted as not applicable. Although some of the measures are applicable, not all of them have automation potential. The measures include Failure Avoidance, Input Validity Check, Undoability, Attractive Interaction needs human interpretation; they are therefore noted as applicable but do not have automation potential. Guceglioglu's measure's implementation took 4 man/hours. While applying the measures some strategies are developed in order to provide consistency with the formulas. The application strategies are summarized in Table 6. Also determination details of the measures that have automation potential are summarized in Table 8. With this case study it is observed that applying the measures on the business processes required time and effort, and the calculations were error-prone in terms of counting the number of arcs and nodes manually.

The approach in this thesis was to automate the measures that have automation potential in order to decrease time and effort spent, and provide more accurate results. The AQM is conducted in to the COSMOS tool which is a Meta-Model Design Editor integrated into the Kama modeling environment. The automation of the measures is implemented by using the COSMOS's local database in order to retrieve necessary data for the calculations. This way the user is enabled to measure the quality of their business processes while modeling them. The details of the automation of the measures are summarized in Table 10.

The second case study was a validation study. After the tool was implemented the Supply Chain Management's AS-IS and TO-BE form processes' quality was measured both manually and automatically. The aim was to compare the time and effort required for measuring in two ways, as well as the accuracy. As a result of the study it is seen that the time required for the manual calculation was 149 minutes where as it could be measured automatically within 16 seconds. In addition, the required effort is certainly much higher than automatically measurement. Since manual calculation can cause inconsistency, in this study the measurement results are compared and seen that in manual calculations there has been some mistakes. Such mistakes are caused by counting the nodes and arcs wrong, which is due to the complexity of the process model. The AQM tool retrieves the count of nodes and arcs from the database and therefore the modeled process's data is retrieved correctly.

As a result of this thesis, quality measures which can give feedback before the execution of the processes are searched in the literature. After identifying the measures, they are applied on a business process model set for observing their applicability, automation potential and required time. This case study showed that applying measures requires significant time and effort. Since process improvement has a continuous approach, quality measurement is not something that will be done by once. As result of this, the AQM tool is developed in order to minimize the required time and effort for application of the measures which have automation potential. Then the second case study is conducted for validating this tool to prove that the measures which have automation potential are really can be automated.

5.2 Future Work

The case studies of these measures are only conducted into Human Resources Management and Supply Chain Management processes. The application can be conducted to more business processes from different sectors. In this way we can continue to evaluate the tool's effectiveness and efficiency. With more case studies, it would also be possible to compare the results and evaluate measures' effectiveness.

As this research area is newly developing, new measures will be adopted or implemented. Therefore, this tool can be expanded adding other frameworks or measures which can be calculated over eEPC diagrams. This way the AQM will be easier to improve, and provide more comprehensive results for the organizations.

This quality measurement automation is conducted in to the COSMOS tool. In other words for measurement calculations it is using COSMOS's database and it is implemented according to this database design. If the other available tool's database design fits to the COSMOS's design our implementation can be prepared as a "plug-in". The business process models, modeled using applications other than COSMOS can be exported as XML type file. Importing this XML type file to the AQM can provide a more general work environment. Our future work is to develop the AQM by integrating an XML type import application.

REFERENCES

Azuma M., Mole, D. Mole.: Software management practice and metrics in the European community and japan: Some results of a survey. Journal of Systems and Software, 26(1):5–18, 1994.

Bilgi Grubu Ltd. Şti., COSMOS: Kurumsal Süreç Modelleme Sistemi Projesi (TEYDEB-3050564) Proje Sonuç Raporu, Mart 2008, Ankara

Cardoso, J.; Mendling, J.; Neuman, G. & Reijers, H.A. (2006). A discourse on complexity of process models. In: Eder, J.; Dustdar, S. et al, editors, BPM 2006 workshops. Lecture Notes in Computer Science 4103, Springer-Verlag, Berlin, pp. 115-126.

Coskuncay, A., Aysolmaz, B., Demirors, O., Bilen, O. & Dogan, I: Bridging the Gap Between Process Modeling and Software Requirements Analysis: A Case Study. In. MCIS 2010 Proceedings.

Ferdian . A Comparison of Event-driven Process Chains and UML Activity Diagram for Denoting Business Processes. Master's thesis (Studienarbeit), Technische Universität Hamburg-Harburg, 2001.

Ghani, A., Muketha, K.T., Wen, W.P. (2008), "Complexity metrics for measuring the understandability and maintainability of business process models using goal-question-metric", International Journal of Computer Science and Network Security, Vol. 8 pp.219-25

Gruhn, V., and Laue, R. (2006). Complexity metrics for business process models. In: Witold Abramowicz and Heinrich C. Mayer, editors, 9th international conference on business information systems (BIS 2006), vol. 85 of Lecture Notes in Informatics, pp. 1-12.

Guceglioglu, S., Demirors, O., van der Aalst, W.M.P.: Using Software Quality Characteristics to Measure Business Process Quality, BPM 2005, LNCS 3649, pp. 374-379, (2005).

Guceglioglu, S.: A Pre-Enactment Model for Measuring Process Quality. Phd Thesis, METU, Ankara, Turkey (2006).

Guceglioglu, S., Demirors O .: The Application of a New Process Quality Measurement Model for Software Process Improvement Initiatives, QSIC,2011

Gurbuz O., Guceglioglu, S., Demirors O.: Application of Process Quality Measurement Frameworks for Human Resource Management Processes, ICQR, Bangkok, Thailand, 2011.

Henry S., Kafura D.: Software structure metrics based on information-flow. IEEE Transactions On Software Engineering, 7(5):510–518, 1981.

ISO/IEC FCD 9126-1.2 :Information Technology, Software Product Quality, Part 1: Quality Model. (2000).

KAMA-C4ISRMOS-YTD3-s00, Software Design Document

Khalikov, F. : COSMOS – META - MODEL EDITOR FOR CONCEPTUAL MODELING, Technical Report, METU, Ankara, Turkey, 2008.

Khlif W., Makni L.: Quality metrics for business process modeling. In WSEAS International Conference on APPLIED COMPUTER SCIENCE (ACS´09). Genova: WSEAS Press, p. 195 - 200, ISBN 978-960-474-127-4, ISSN 1790-5109 (2009).

McCabe T. J.. A complexity measure. IEEE Transactions on Software Engineering, 2(4):308–320, 1976.

Mendling, J.: Testing Density as a Complexity Metric for EPCs. Technical Report JM-2006-11-15. Vienna University of Economics and Business Administration. Retrieved from http://wi.wu-wien.ac.at/home/mendling/publications/TR06-density.pdf

Muketha, G.M., A.A.A. Ghani, M.H. Selamat and R. Atan, 2010. A survey of business process complexity metrics. Inform. Technol. J., 9: 1336-1344

Ralph, F.S., (2007). Business process management and balanced scorecard: using processes as strategic drivers. Wiley & Sons.

Vanderfeesten, I., Cardoso, J., Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Quality Metrics for Business Process Models. In: Fischer, L. (ed.) BPM and Workflow Handbook 2007, Future Strategies, USA, May 2007, pp. 179–190 (2007a).

Vanderfeesten.I, Cardoso.J, Reijers.H.A: A weighted coupling metric for business process models, Technische Universities Eindhoven, Department of Technology Management, PO Box 513, 5600 MB Eindhoven, The Netherlands, University of Madeira, Department of Mathematics and Engineering, 2007b.

Vanderfeesten, I., Reijers, H.A., Mendling, J., van der Aalst, W.M.P., Cardoso, J.: On a quest for good process models: the cross-connectivity metric, CAiSE'08: Proceedings of the 20th International Conference on Advanced Information Systems Engineering, pp480-94 (2008).

APPENDICES

APPENDIX A: The codes of the algorithms defined in the approach

Table 25. Density

```
SqlCommand arcs = new SqlCommand("select COUNT (*) from Relations where DiagramID = (select ID
from Diagrams where ModelID = ( select ID from Models where Name= '" + textBox18.Text + "')) and
Name='Rect Relation Arrow'", yeni);
        int arcsayi = int.Parse(arcs.ExecuteScalar().ToString());
        textBox19.Text = arcsayi.ToString();

        int amin = int.Parse(node.ExecuteScalar().ToString()) - 1;

        if (c != 1)
        {
          if (c % 2 == 0)
          {
            float cmaxeven = ((c / 2) + 1) * ((c / 2) + 1);


            textBox6.Text = ((arcsayi - amin) / (cmaxeven + 2 * (ev + f + pi) - amin)).ToString();
          }
          else
          {
            float cmaxodd = (((c - 1) / 2) + 1) * (((c - 1) / 2) + 1) + ((c - 1) / 2) + 1;

            textBox6.Text = ((arcsayi - amin) / (cmaxodd + 2 * (ev + f + pi) - amin)).ToString();
          }
        }
        else
        {
          textBox6.Text = "0";

        }
```

Table 26. Weighted Coupling

```
SqlCommand weightedc = new SqlCommand("select   m.Name, SourceModelElementID, m2.Name,
TargetModelElementID as Targete from Relations r,DiagramsToModelElements
d,DiagramsToModelElements d2, ModelElements m, ModelElements m2 where
SourceModelElementID=d.ID and TargetModelElementID=d2.ID  and d.ModelElementID=m.ID and
d2.ModelElementID=m2.ID and m.ModelID= (select ID from Models where Name ='" + textBox18.Text
+ "') and SourceModelElementID in (select ID from DiagramsToModelElements where ModelElementID
in (select ID from ModelElements where Name='Function' or Name='Event' or Name='OR' or
Name='ProcessInterface' or Name='XOR' or Name='AND')) and TargetModelElementID in (select ID
from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where
Name='Function' or Name='Event' or Name='OR' or Name='ProcessInterface'or Name='XOR' or
Name='AND' or Name='Sonlanma Durumu' ))", yeni);

        SqlDataReader read = weightedc.ExecuteReader();

        string[,] names= new string [50,4];
        float wcoupling = 0;

        int counter = 0;
        float fout,fin;

        while (read.Read())
        {
            names[counter,0] = read[0].ToString();
            names[counter,1] = read[1].ToString();
            names[counter, 2] = read[2].ToString();
            names[counter, 3] = read[3].ToString();

            counter++;
        }
        for (int i = 0; i < counter; i++)
        {

            if (names[i, 0] == "Function" || names[i, 0] == "Event" || names[i, 0] == "ProcessInterface")
            {
                if (names[i, 2] == "Function" || names[i, 2] == "Event" || names[i, 2] == "ProcessInterface")
                {
                    wcoupling++;
                }
                if (names[i, 2] == "OR")
                {
                    fout=count(names, names[i, 1],counter,1);
                    fin = count(names, names[i, 1],counter,3);

                    if (((ikiUzeri(fin) - 1) * (ikiUzeri(fout) - 1)) != 0)
                    {

                        wcoupling += fout * ((1 / ((ikiUzeri(fin) - 1) * (ikiUzeri(fout) - 1))) + ((((ikiUzeri(fin)
- 1) * (ikiUzeri(fout) - 1))) - 1) / (((ikiUzeri(fin) - 1) * (ikiUzeri(fout) - 1))) * (1 / (fin * fout)));
                    }
                }
                if (names[i, 2] == "XOR")
                {
```

Table 26 (cont.)

```
                    fout=count(names, names[i, 1],counter,1);
                    fin = count(names, names[i, 1],counter,3);
                    if (fout * fin != 0)
                    {
                        wcoupling += fout * (1 / (fout * fin));
                    }
                }
                if (names[i, 2] == "AND")
                {
                    fout = count(names, names[i, 1], counter,1);

                    wcoupling += fout ;
                }


            }

        }

        yeni.Close();
        yeni.Open();
        SqlCommand T = new SqlCommand("select COUNT(*) from ModelElements where ModelID=
(select ID from Models where Name='" + textBox18.Text + "') and (Name='Function' or Name='Event' or
Name='ProcessInterface')", yeni);
        int to = int.Parse(T.ExecuteScalar().ToString());

        textBox5.Text = (wcoupling / (to*(to-1))).ToString();
```

Table 27. Control Flow Complexity

```
SqlCommand OrOut = new SqlCommand("select COUNT(*) as cnt from Relations r,
DiagramsToModelElements d, ModelElements me, Models m where r.DiagramID=d.DiagramID and
r.SourceModelElementID=d.ID and d.ModelElementID=me.ID and me.Name= 'OR' and
me.ModelID=m.ID and m.Name='" + textBox18.Text + "' group by SourceModelElementID", yeni);

        SqlDataReader reader = OrOut.ExecuteReader();


        int or,x=1;

        int cfc=0;

        string[] array= new string[2];


        while (reader.Read())
        {

            array[0] = reader[0].ToString();
```

Table 27 (cont.)

```
                    or = int.Parse(array[0]);

                    cfc += (ikiUzeri(or) - 1);
                    x = 1;


            }


            yeni.Close();
            SqlCommand XOrOut = new SqlCommand("select COUNT(*) from Relations r, 
DiagramsToModelElements d, ModelElements me, Models m where r.DiagramID=d.DiagramID and 
r.SourceModelElementID=d.ID and d.ModelElementID=me.ID and me.Name= 'XOR' and 
me.ModelID=m.ID and m.Name='" + textBox18.Text + "' group by SourceModelElementID", yeni);
            yeni.Open();
            SqlDataReader reader2= XOrOut.ExecuteReader();

            string holder;
            int xor;

            while(reader2.Read())
            {
               holder = reader2[0].ToString();

               xor = int.Parse(holder);

               cfc += xor;

            }
            yeni.Close();
            yeni.Open();
            SqlCommand AndOut = new SqlCommand("select count(*) from ModelElements where 
Name='AND' and ModelID = (select ID from Models where Name='" + textBox18.Text + "')", yeni);


            int and = int.Parse(AndOut.ExecuteScalar().ToString());

            cfc += and;

            textBox7.Text = cfc.ToString();
```

Table 28. Imported Coupling of a Process (ICP)

```
SqlCommand icp = new SqlCommand("select COUNT(*) from Relations r, DiagramsToModelElements d, ModelElements me, Models m where r.DiagramID=d.DiagramID and r.SourceModelElementID=d.ID and d.ModelElementID=me.ID and (me.Name= 'Event' or me.Name= 'Function' or me.Name= 'ProcessInterface') me.ModelID=m.ID and r.Name='Rect Relation Arrow' and m.Name='" + textBox18.Text + "'", yeni);


        int icpp = int.Parse(icp.ExecuteScalar().ToString());
        textBox8.Text = icpp.ToString();
```

Table 29. Exported Coupling of a Process (ECP)

```
        SqlCommand ecp = new SqlCommand("select COUNT (*) from Relations r, DiagramsToModelElements d, ModelElements me, Models m where r.DiagramID=d.DiagramID and r.TargetModelElementID=d.ID and d.ModelElementID=me.ID and (me.Name= 'Event' or me.Name= 'Function' or me.Name= 'ProcessInterface') and me.ModelID=m.ID and r.Name='Rect Relation Arrow' and m.Name='" + textBox18.Text + "'", yeni);



        int ecpp = int.Parse(ecp1.ExecuteScalar().ToString());
        textBox9.Text = ecpp.ToString();
```

Table 30. Response for Process Coupling (RFP)

```
float rfp=0;

        for (int i = 0; i < counter; i++)
        {
           if (names[i, 0] == "Function" || names[i, 0] == "Event" || names[i, 0] == "ProcessInterface")
           {
              if (names[i, 2] == "XOR" || names[i, 2] == "Function" || names[i, 2] == "Event" || names[i, 2] == "ProcessInterface")
              {
                 rfp+=2;
              }
              if (names[i, 2] == "OR" || names[i, 2] == "AND")
              {
                fout = count(names, names[i, 1], counter,1);

                 rfp +=  fout+ 1;
              }
              if (names[i, 2] == "Sonlanma Durumu")
              {
                 rfp += 1;
              }
           }          }          textBox10.Text = rfp.ToString();
```

Table 31. Locality of Data-Based (LDA)

```
SqlCommand lda = new SqlCommand("select SourceModelElementID, TargetModelElementID from Relations where SourceModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='InputOutput')) and TargetModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='Function')) and DiagramID = (select ID from Diagrams where ModelID =( select ID from Models where Name='" + textBox18.Text + "')) union select SourceModelElementID, TargetModelElementID from Relations where SourceModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='Function')) and TargetModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='InputOutput')) and DiagramID = (select ID from Diagrams where ModelID =( select ID from Models where Name='" + textBox18.Text + "'))", yeni);
            SqlDataReader read3 = lda.ExecuteReader();

            string[,] inout = new string[100, 2];
           float ldaa = 0;

            int ct = 0;
            float y, z;


            while (read3.Read())
            {
               inout[ct, 0] = read3[0].ToString();
               inout[ct, 1] = read3[1].ToString();

               ct++;
            }
            string[] temp = new string[100];

            for (int i = 0; i < ct; i++)
            {
               if(isFunction(names,inout[i,0], counter) && iscounted(temp, inout[i,0]))
               {

               y=count(inout,inout[i,0],ct,0);
               z = count(inout, inout[i, 0], ct,1);
                ldaa+=(z / (y +z));
                temp[i] = inout[i, 0];
               }
            }
            textBox11.Text = ldaa.ToString();
```

Table 32. Complexity

```
        SqlCommand totalactivity = new SqlCommand("select COUNT(*) from ModelElements where
(Name='Function' or Name='ProcessInterface')and ModelID= (select ID from Models where Name='" +
textBox18.Text + "')", yeni);
        float activity = int.Parse(totalactivity.ExecuteScalar().ToString());

        SqlCommand complexity = new SqlCommand("select COUNT(*) from Relations where
SourceModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select
ID from ModelElements where Name='Function' )) and TargetModelElementID in (select ID from
DiagramsToModelElements where ModelElementID in (select ID from ModelElements where
Name='OR'or Name='XOR' or Name='AND')) and DiagramID = (select ID from Diagrams where
ModelID =( select ID from Models where Name='" + textBox18.Text + "'))", yeni);
        float comp = int.Parse(complexity.ExecuteScalar().ToString());
        textBox12.Text = (1 - (comp / activity)).ToString();
```

Table 33. Coupling

```
SqlCommand coupling = new SqlCommand("select COUNT(*) from ModelElements where
Name='ProcessInterface' and ModelID= (select ID from Models where Name='" + textBox18.Text + "')",
yeni);
        float coup = int.Parse(coupling.ExecuteScalar().ToString());
        textBox13.Text = (1 - (coup / activity)).ToString();
```

Table 34. Restorability

```
SqlCommand restorability = new SqlCommand("select count (distinct (SourceModelElementID)) from
Relations where SourceModelElementID in (select ID from DiagramsToModelElements where
ModelElementID in (select ID from ModelElements where Name='Function' or Name='ProcessInterface'
)) and TargetModelElementID in (select ID from DiagramsToModelElements where ModelElementID in
(select ID from ModelElements where Name='InputOutput' )) and DiagramID = (select ID from Diagrams
where ModelID =( select ID from Models where Name='" + textBox18.Text + "'))", yeni);
        float rest = int.Parse(restorability.ExecuteScalar().ToString());
        textBox14.Text = (rest / activity).ToString();
```

Table 35. Restoration Effectiveness

```
SqlCommand restoration = new SqlCommand("select distinct(SourceModelElementID) from Relations where SourceModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='Function' or Name='ProcessInterface' )) and TargetModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where  Name='InputOutput')) and DiagramID = (select ID from Diagrams where ModelID =( select ID from Models where Name='" + textBox18.Text + "'))", yeni);
        SqlDataReader readd = restoration.ExecuteReader();
        string[] sources = new string[100];
        int ind=0;
        while(readd.Read())
        {
          sources[ind]=readd[0].ToString();
          ind++;
        }
        yeni.Close();
        yeni.Open();

        SqlCommand clusterr = new SqlCommand("select SourceModelElementID from Relations where SourceModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='Function' or Name='ProcessInterface' )) and TargetModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where  Name='cluster')) and DiagramID = (select ID from Diagrams where ModelID =( select ID from Models where Name='" + textBox18.Text + "'))", yeni);
        SqlDataReader cread = clusterr.ExecuteReader();
        string[] csource = new string[100];
        int ind2 = 0;
        while (cread.Read())
        {
          csource[ind2] = cread[0].ToString();
          ind2++;
        }
        yeni.Close();
        yeni.Open();

       float result = isIn(sources, csource, ind2, ind);
       textBox15.Text = (result / rest).ToString();
```

Table 36. IT Usage (ITU)

```
SqlCommand ITU = new SqlCommand("select COUNT(*) from ModelElements where Name='cluster' and ModelID= (select ID from Models where Name='" + textBox18.Text + "')", yeni);
        float itu = int.Parse(ITU.ExecuteScalar().ToString());
        textBox16.Text = (itu / activity).ToString();
```

Table 37. IT Denstiy (ITD)

```
SqlCommand ITD = new SqlCommand("select TargetModelElementID from Relations where SourceModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='InputOutput')) and TargetModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='Event' or Name='Function' or Name='ProcessInterface')) and DiagramID = (select ID from Diagrams where ModelID =( select ID from Models where Name='" + textBox18.Text + "')) union select SourceModelElementID from Relations where SourceModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='Event' or Name='Function' or Name='ProcessInterface')) and TargetModelElementID in (select ID from DiagramsToModelElements where ModelElementID in (select ID from ModelElements where Name='InputOutput' )) and DiagramID = (select ID from Diagrams where ModelID =( select ID from Models where Name='" + textBox18.Text + "'))", yeni);
        SqlDataReader ritd = ITD.ExecuteReader();
        string[] itds = new string[100];
        int countt = 0;
        while (ritd.Read())
        {
           itds[countt] = ritd[0].ToString();
           countt++;
        }
        textBox17.Text = (itu / countt).ToString();
        yeni.Close();
        yeni.Open();
```

APPENDIX B: Human Resources Management Improvement List

Processes	Total	FAD	AA	R	FA	CP	CA	FC
HR Planning	6	2	1	1	1			
Employment	9	2				1	2	2
Ordering	12	7				1	1	
Training	6				1	1	1	1
Payed and Non Payed Leaves	2					1		
HR Performance Evaluation	7	1	1					1
Ending Employment Contract	2		1				1	
Managing Announcements	1							
Work Health and Security, Work Ethichs, Employee Pleasure	4		1					

APPENDIX C: The Supply Chain Management processes modeled in the COSMOS Tool

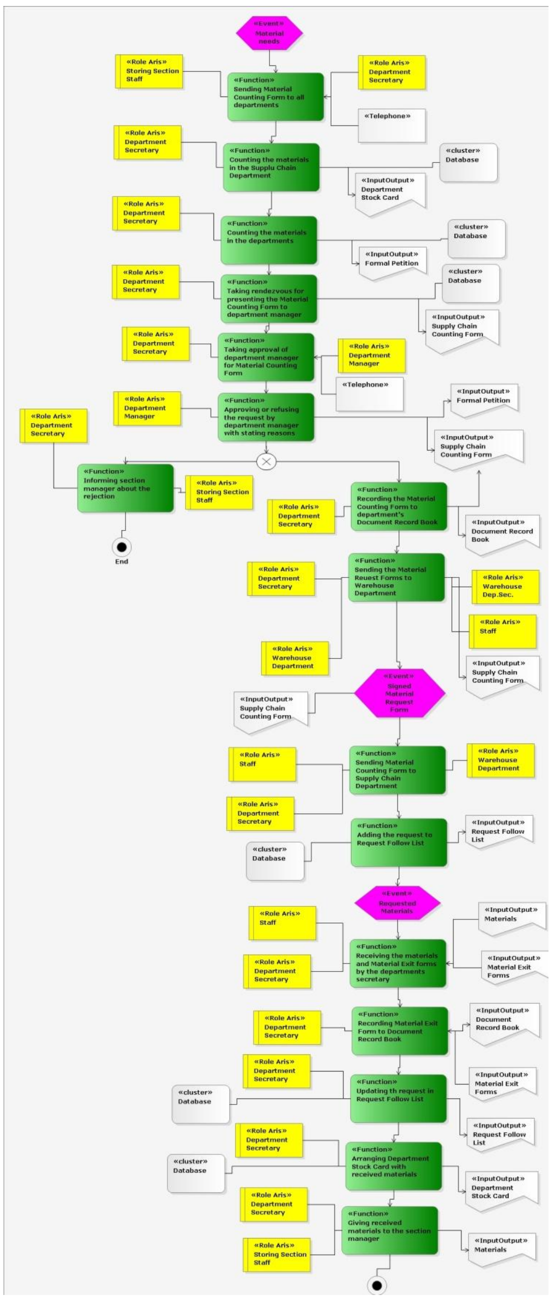


Figure 27. Material Request AS-IS

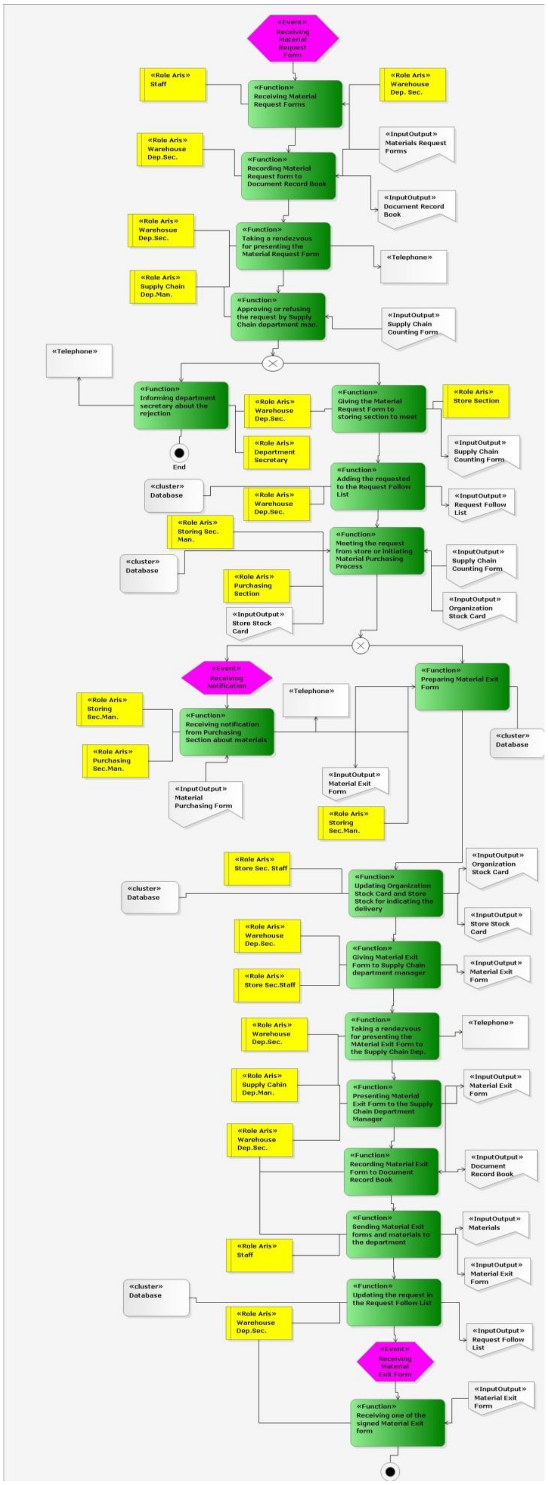


Figure 28. Meeting Material Request AS-IS

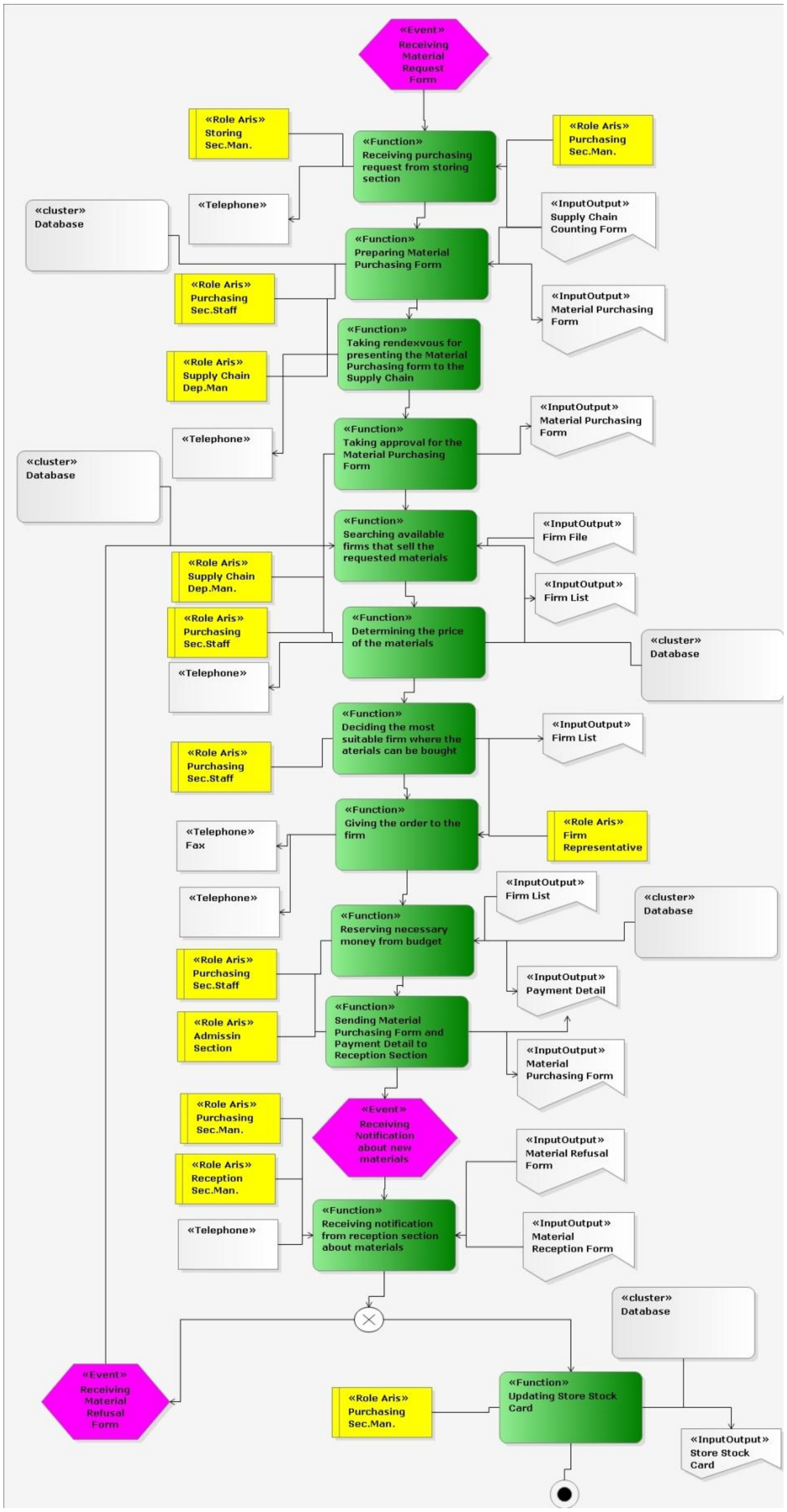


Figure 29. Material Purchasing AS-IS

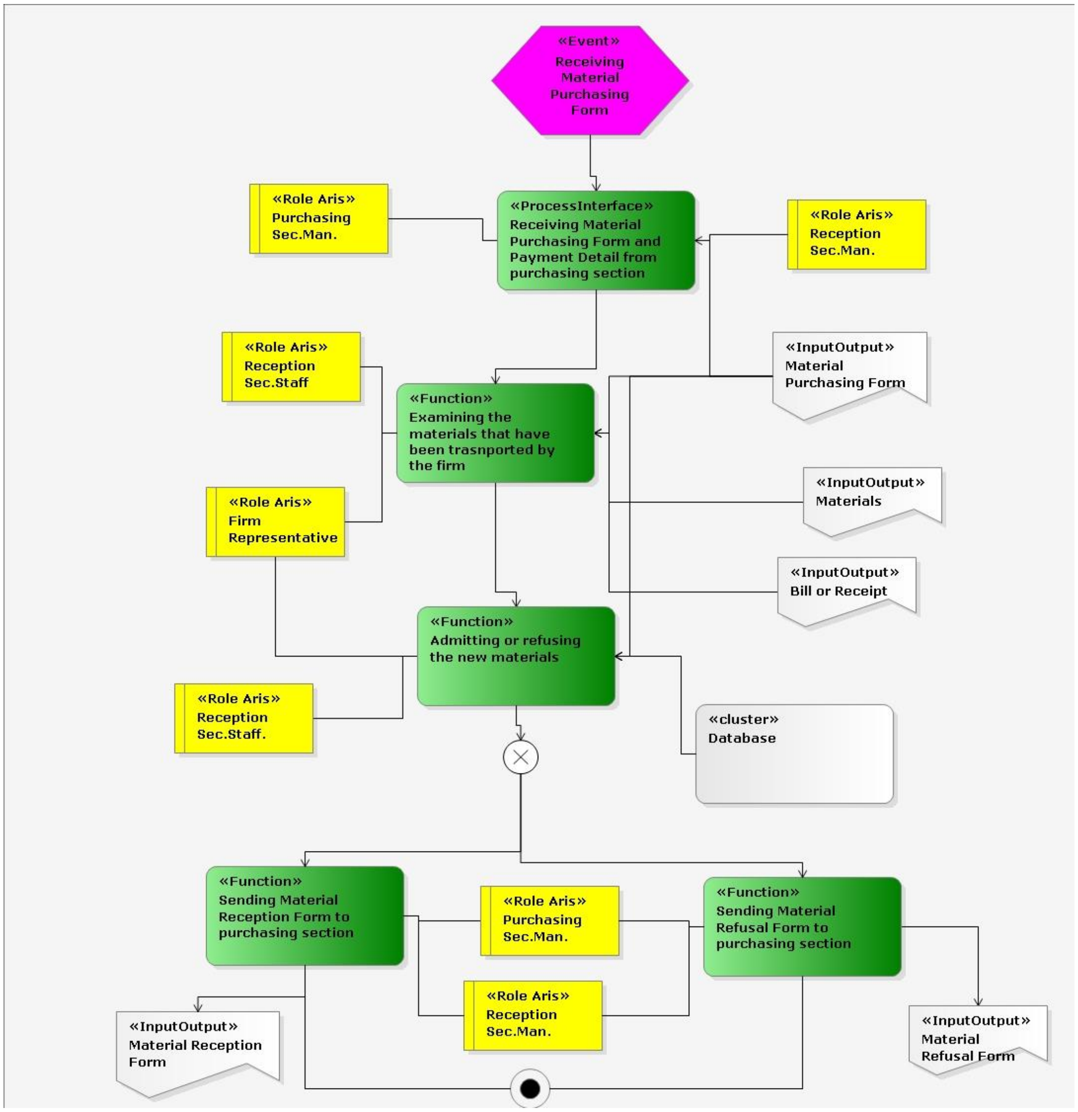


Figure 30. Material Registration

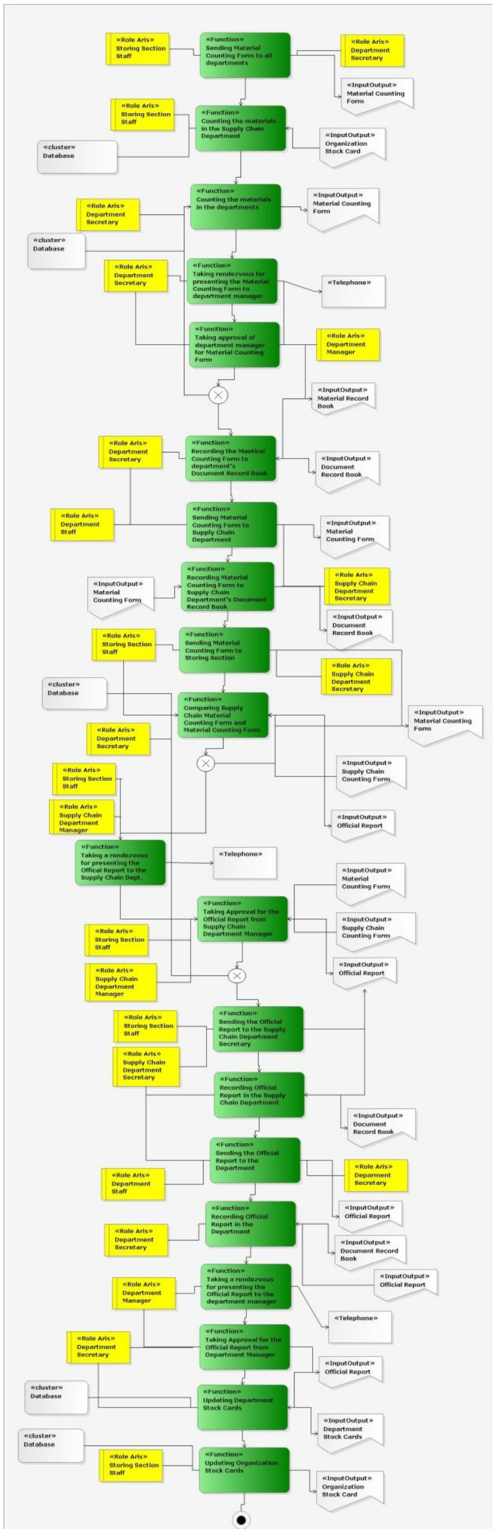


Figure 31. Material Counting AS-IS

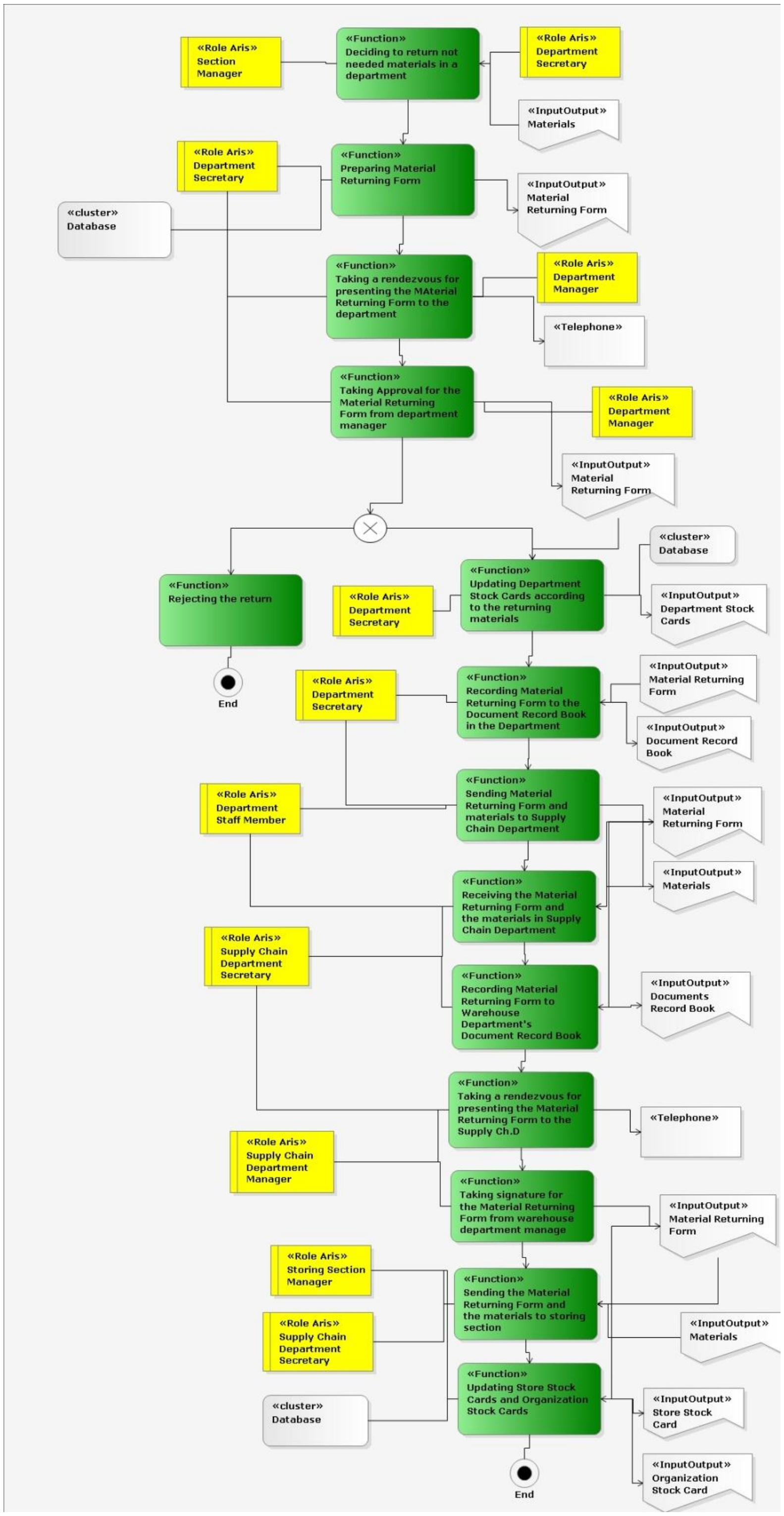


Figure 32. Material Returning AS-IS

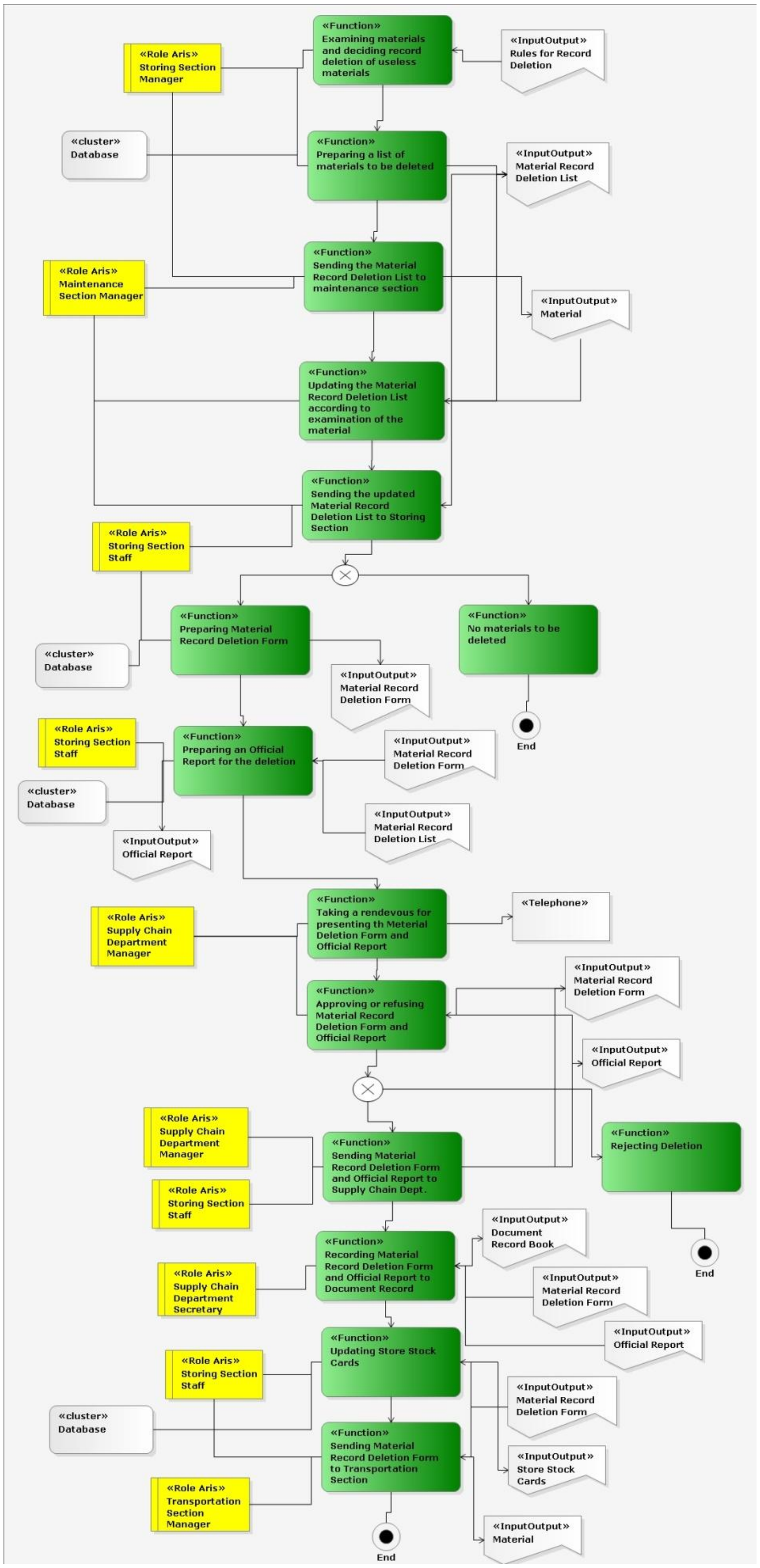


Figure 33. Material Record Deletion AS-IS

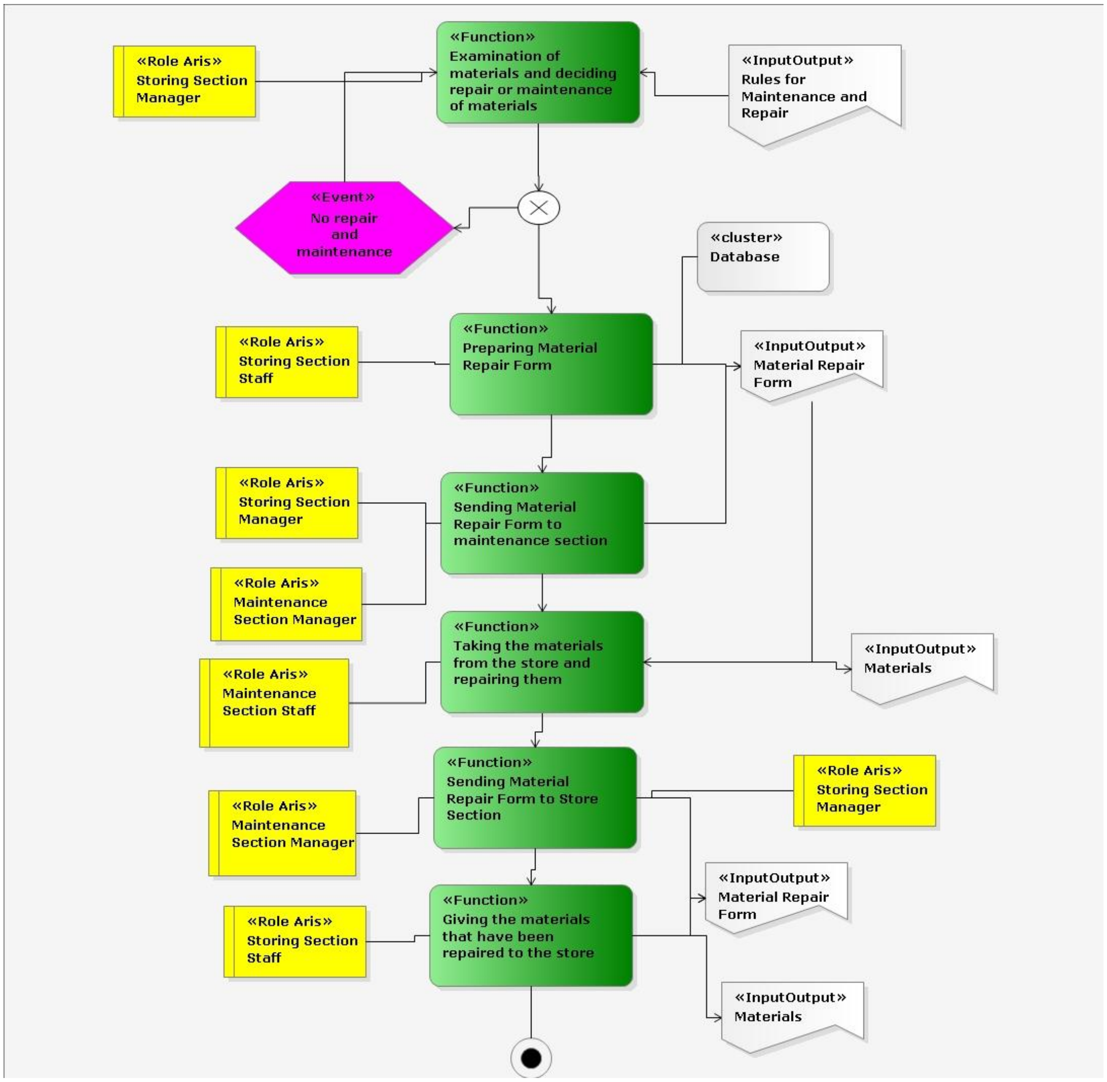


Figure 34. Material Repair and Maintenance AS-IS

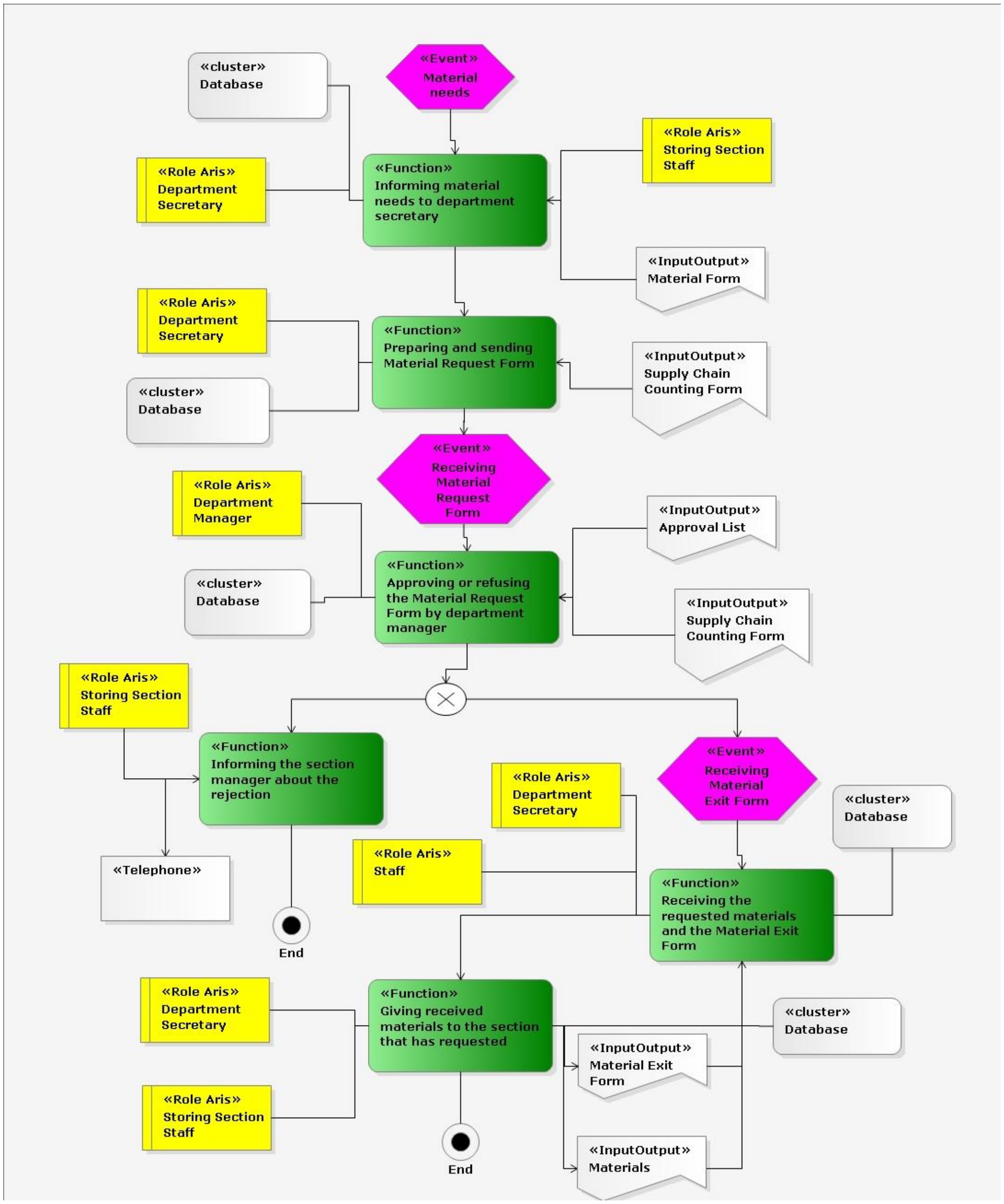


Figure 35. Material Request TO-BE

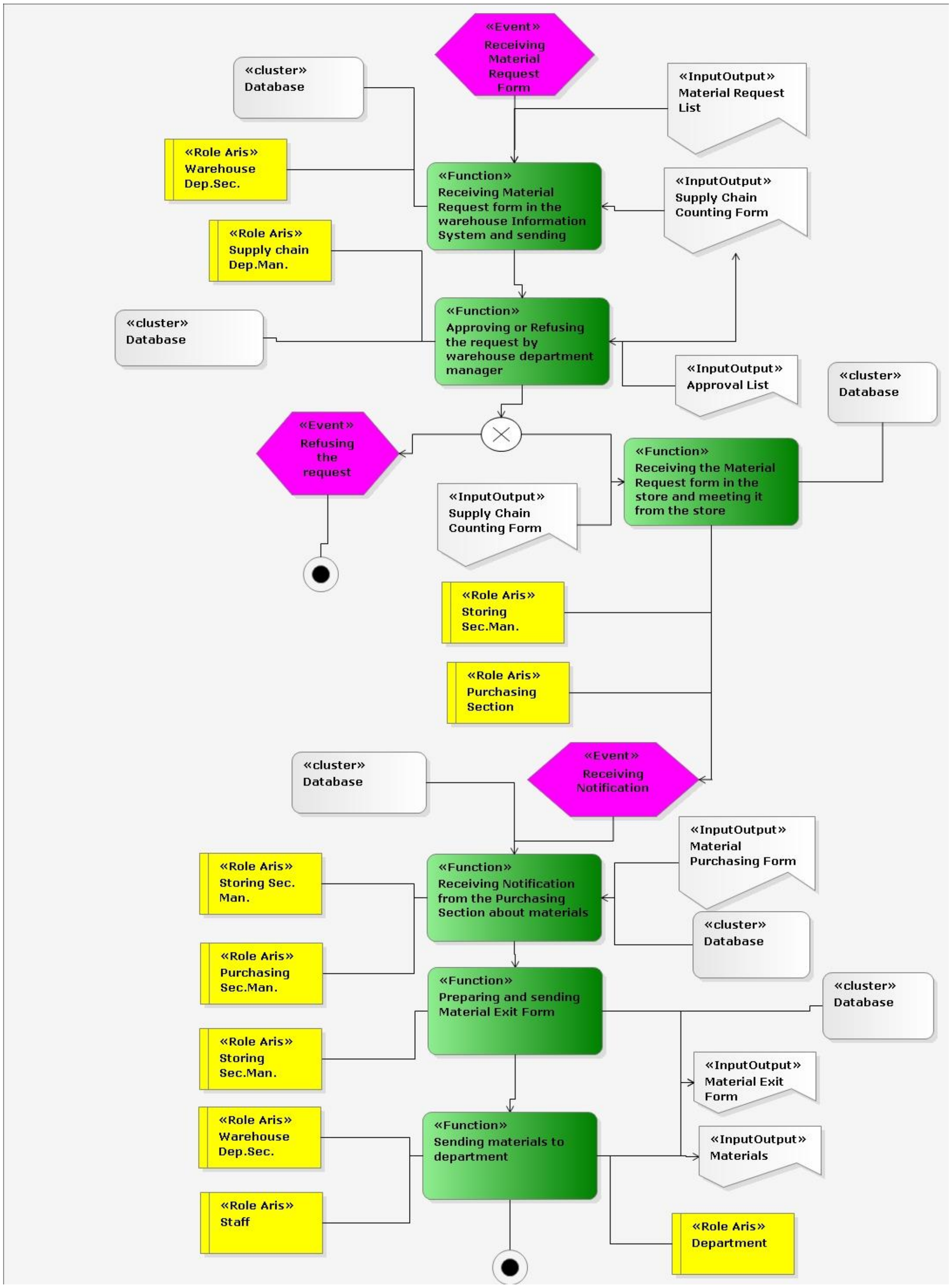


Figure 36. Meeting Material Request TO-BE

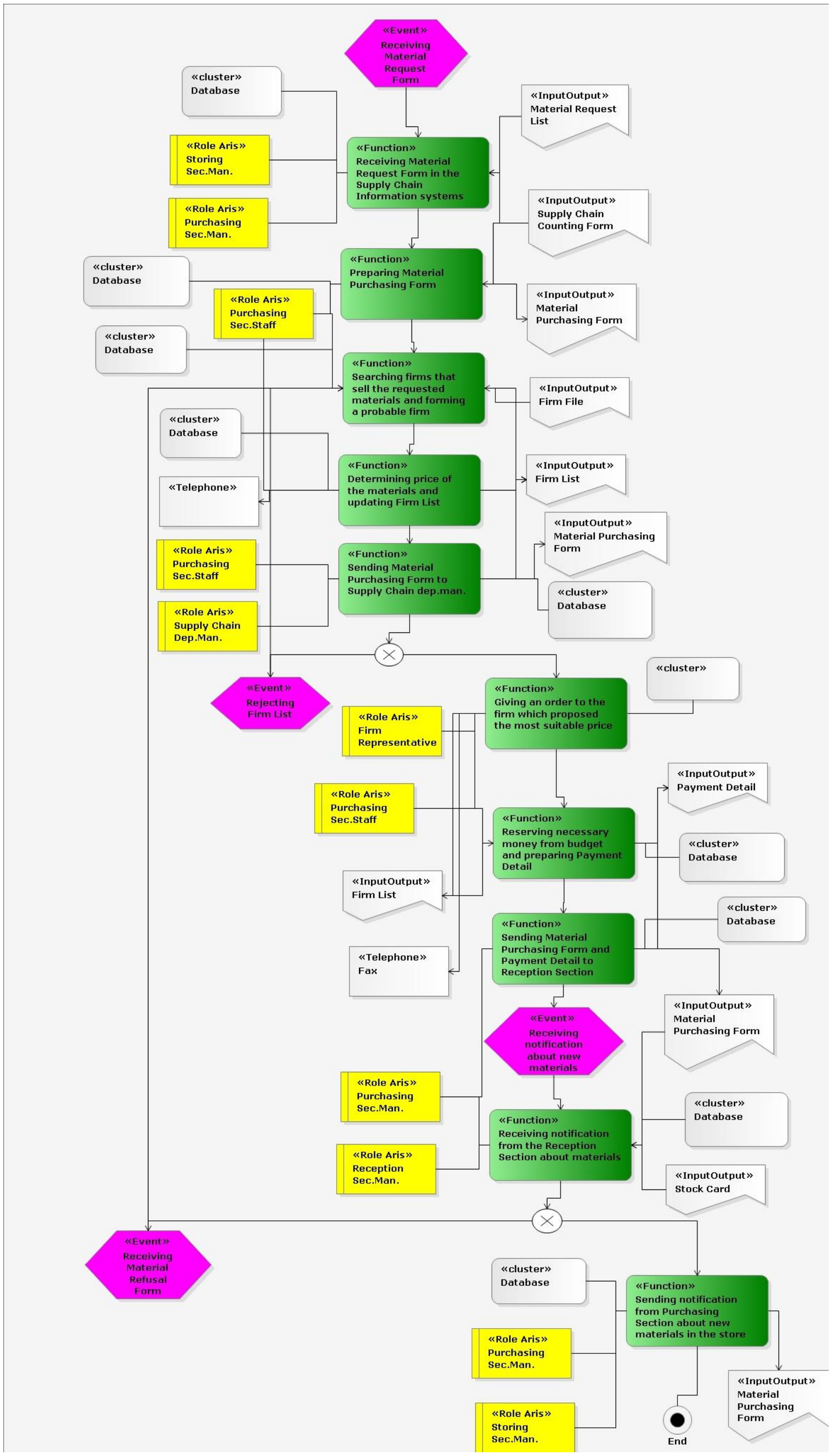


Figure 37. Material Purchasing TO-BE

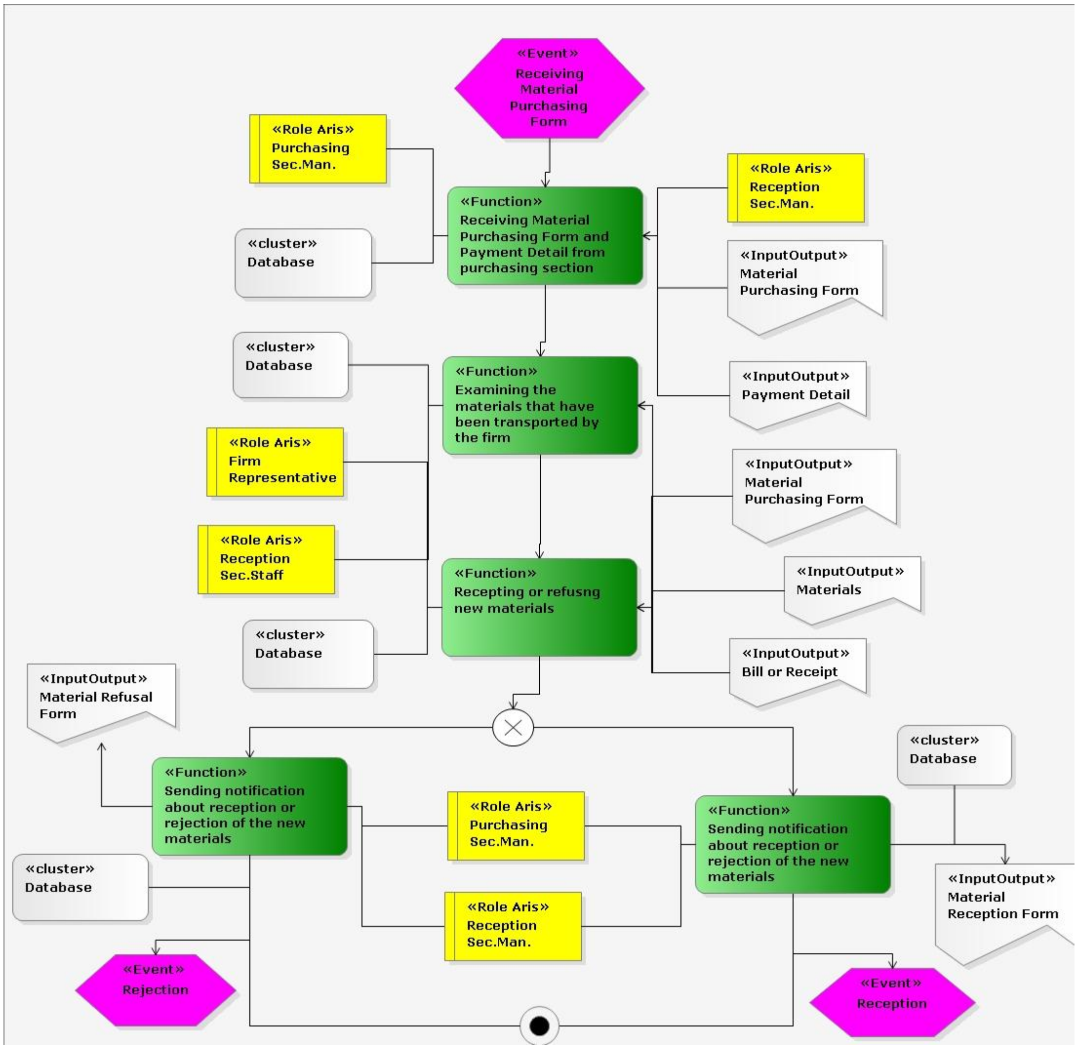


Figure 38. Material Registration TO-BE

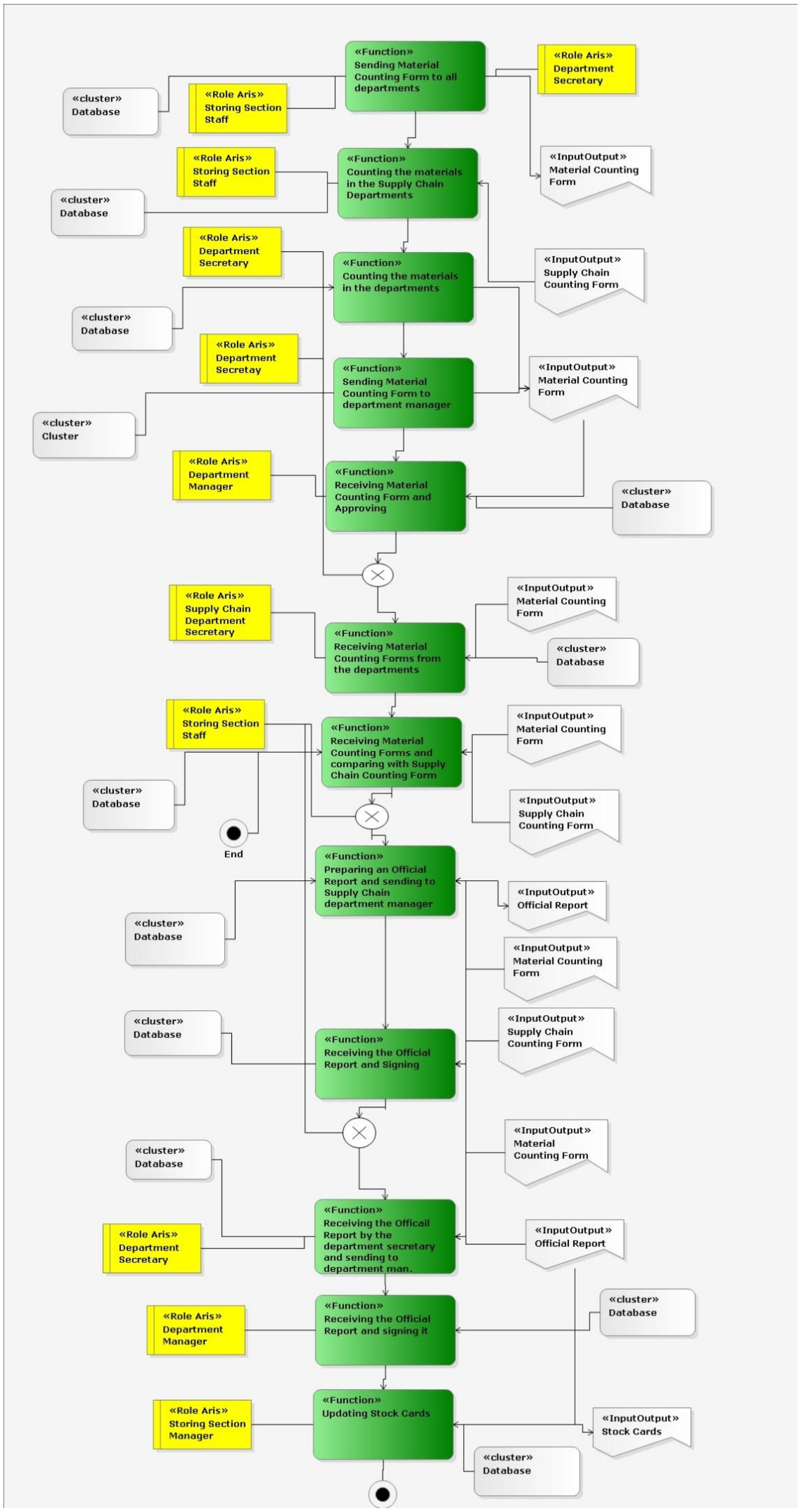


Figure 39. Material Counting TO-BE

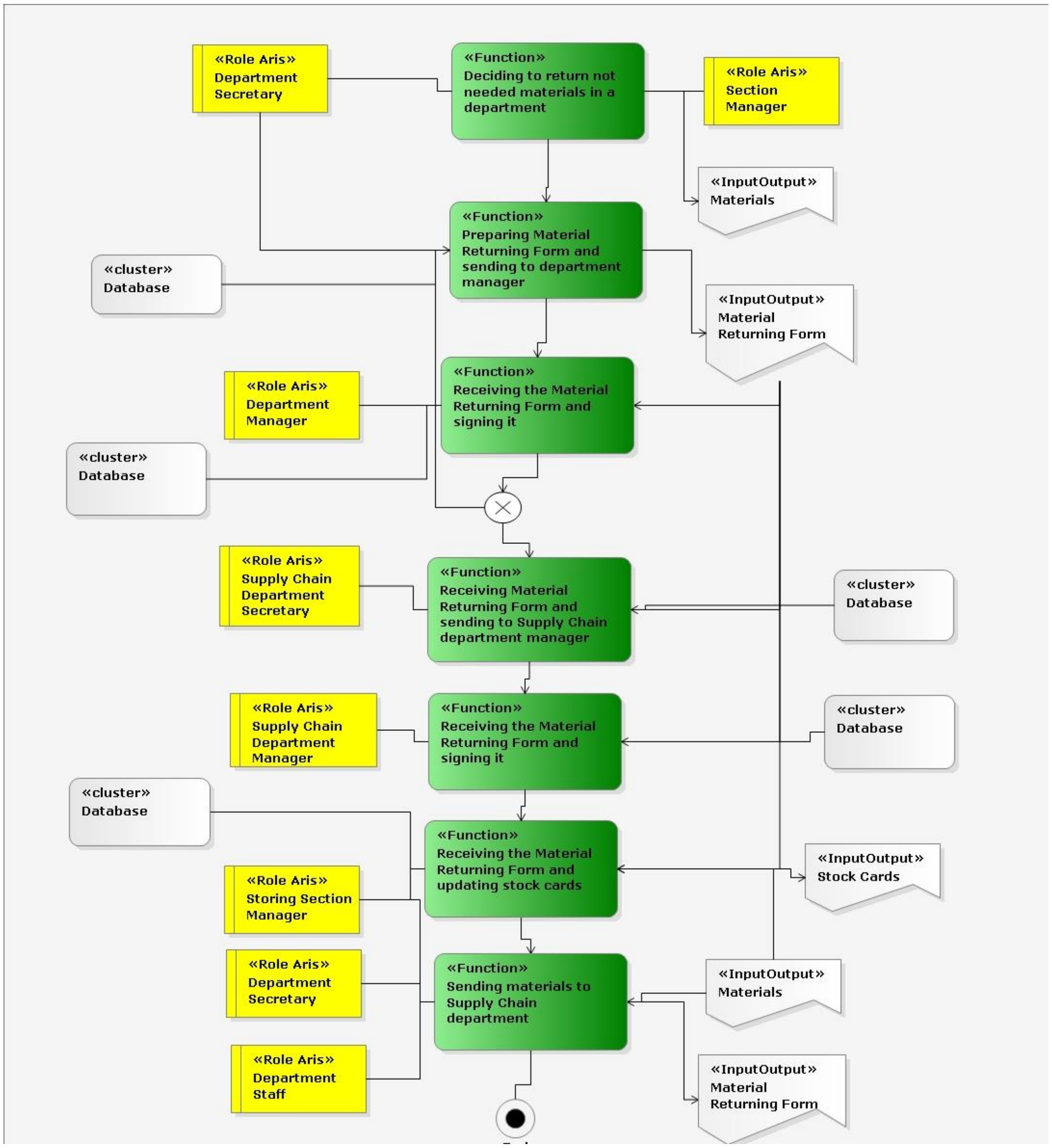


Figure 40. Material Returning TO-BE

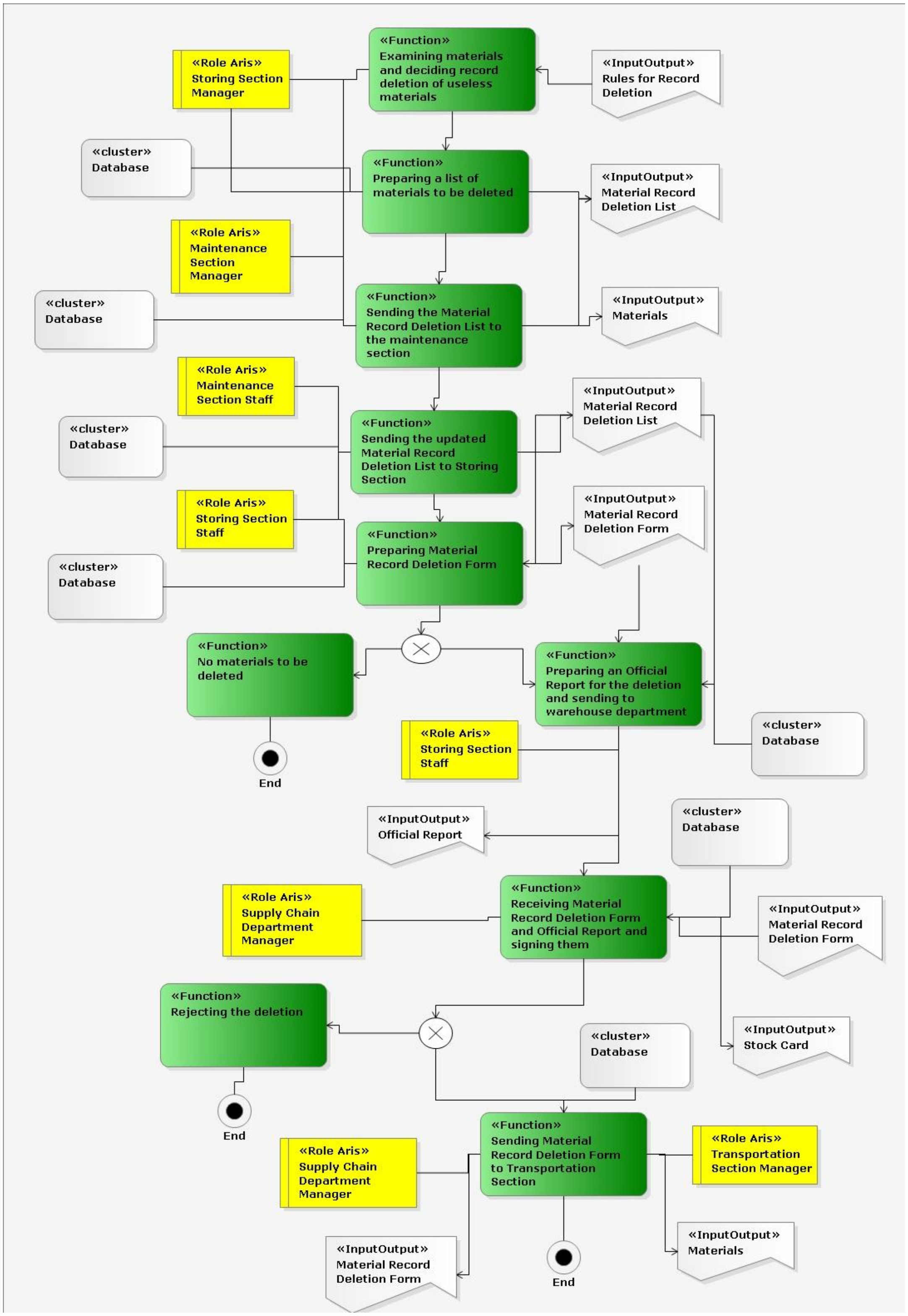


Figure 41. Material Record Deletion TOBE

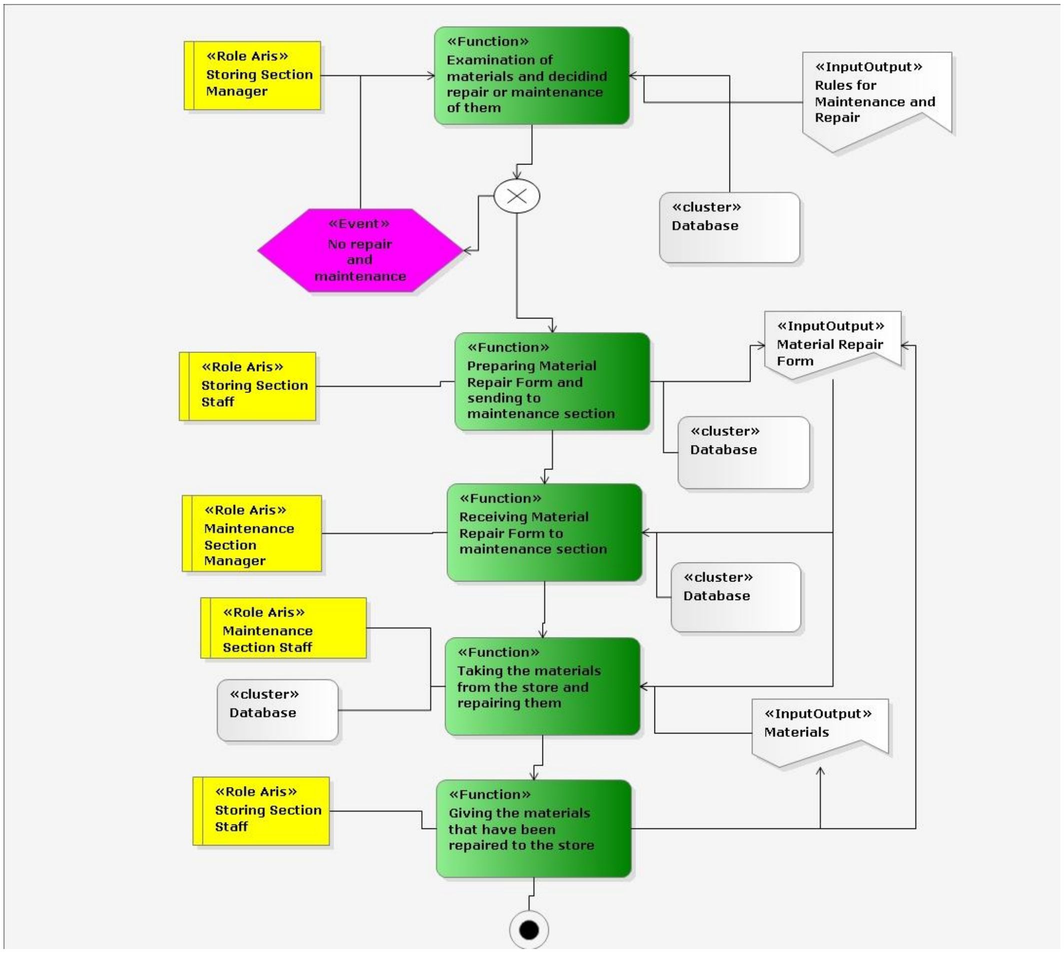


Figure 42. Material Repair and Maintenance TOBE