UPROM: A UNIFIED BUSINESS PROCESS MODELING METHODOLOGY

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

BY

BANU AYSOLMAZ

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

JUNE 2014

UPROM: A UNIFIED BUSINESS PROCESS MODELING METHODOLOGY

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

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

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

Prof. Dr. Onur Demirörs ____________________
Supervisor, **Information Systems, METU**

Examining Committee Members:

Prof. Dr. Semih Bilgen ____________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Onur Demirörs ____________________
Information Systems, METU

Assoc. Prof. Dr. Aysu Betin Can ____________________
Information Systems, METU

Assoc. Prof. Dr. Vahid Garousi ____________________
Dept. of Electrical and Computer Engineering, University of Calgary

Assoc. Prof. Dr. Altan Koçyiğit ____________________
Information Systems, METU

Date: 10.06.2014

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

Name, Last Name : Banu Aysolmaz

Signature : ____________________

ABSTRACT

UPROM: A UNIFIED BUSINESS PROCESS MODELING METHODOLOGY

Aysolmaz, Banu

Ph. D., Department of Information Systems

Supervisor: Prof. Dr. Onur Demirörs

June 2014, 207 pages

The information captured in business process models can be utilized for many artifacts in the organizations. However, relations between the models and related artifacts are usually either not established or only partially formed. This results in increased effort, consistency, completeness and maintainability problems regarding the artifacts. This study proposes a unified business process modeling methodology, UPROM to integrate analysis for process documentation and automation so that related artifacts can be generated in a complete, consistent and traceable way. UPROM comprises notation, meta-model, process, guidelines and artifact generation procedures. A prototype tool is developed to support UPROM. Application of UPROM enables users to automatically generate the artifacts of user requirements document, COSMIC software size estimation and process metrics list for process automation software; and process definition document, business glossary and process improvement list for process documentation. A multiple case study is conducted to evaluate the application of the methodology, generate the artifacts and validate the results.

Keywords: Business process modeling, requirements analysis, COSMIC software size estimation, process metrics, process documentation

ÖZ

UPROM: BÜTÜNLEŞİK İŞ SÜREÇLERİ MODELLEME YÖNTEMİ

Aysolmaz, Banu

Doktora, Bilişim Sistemleri

Tez Yöneticisi: Prof. Dr. Onur Demirörs

Haziran 2014, 207 sayfa

İş süreç modelleri ile toplanan bilgi, kurumlarda başka birçok iş ürünü için de kullanılabilir. Ancak genelde modellerle ilgili iş ürünleri arasındaki bağlantı ya hiç kurulmamakta, ya da kısmen oluşturulmaktadır. Bu da iş ürünleri için artan işgücü ve tutarlılık, tamlık ve bakım sorunlarına neden olmaktadır. Bu çalışma UPROM isimli bir bütünleşik iş süreçleri modelleme yöntemi önermektedir. Bu yöntem süreç otomasyonu ve dokümantasyonu için analiz faaliyetlerini birleştirerek ilgili iş ürünlerinin tam, tutarlı ve izlenebilir şekilde oluşturulmasını amaçlamaktadır. UPROM notasyon, meta-model, süreç, kılavuz ve iş ürünü oluşturma prosedürlerinden oluşmaktadır. UPROM'u desteklemek için prototip bir araç geliştirilmiştir. UPROM'u uygulayarak kullanıcılar süreç otomasyon yazılımı için kullanıcı gereksinimleri dokümanı, COSMIC yazılım büyüklük kestirimi ve süreç metrikleri listesi; süreç dokümantasyonu için süreç tanımlama dokümanı, iş sözlüğü ve süreç iyileştirme listesi oluşturabilirler. Yöntemin uygulanmasını değerlendirmek, iş ürünlerini oluşturmak ve sonuçları doğrulamak için çoklu durum çalışması uygulanmıştır.

Anahtar Kelimeler: İş süreçleri modelleme, gereksinim analizi, COSMIC yazılım büyüklük kestirimi, süreç metrikleri, süreç dokümantasyonu

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor Dr. Onur Demirörs for his trust in my studies, always coming up with new ideas, encouraging me in hard times and never over-pressurizing. I would also like to thank my committee members Dr. Semih Bilgen and Dr. Altan Koçyiğit for their encouragement and support through the thesis studies.

Many professionals collaborated with me in the case studies. My studies would be inconclusive without their insights and support. I want to thank the project group of Turksat, officials from Company Registrar's Office of TRNC, and process analysts group of METU Computer Center.

My studies would not be complete without the collaboration of colleagues and dear friends from Software Management Research Group and Bilgi Grubu and their valuable company. I want to present my sincere thanks to Erdir Ungan, Özden Özcan Top, Gökçen Yılmaz, Ahmet Coşkunçay, Özge Gürbüz, Murat Salmanoğlu, Ebru Gökalp and other friends supporting me. I will never forget the long hours, good and bad moods we experienced together with Özden since the start of the PhD; and our time together in the toughest periods and precious guidance provided by Filiz.

I am grateful for the continuous support of the Informatics Institute personnel. I especially want to thank Sibel Gülnar and Ali Kantar for their support and friendship since the start of my graduate studies.

I am very lucky to have Deniz as the best companion in all ways, and to be together at all moments while striving through PhD stages. Most of the times we have not only been the only persons to motivate one another, put the mind back on track and provide the best research advisory; but also to find the most enjoyable ways to defer the studies and experience many other aspects of life.

Lastly, I would like to thank my family; my mother, father and brother for their continuous support. Although they became tired of my never ending studies, I know they trusted that I will complete the thesis in the best possible way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BAM	: Business Activity Monitoring
BP	: Business Process
BPA	: Business Process Analysis
BPEL	: Business Process Execution Language
BPI	: Business Process Improvement
BPM	: Business Process Management
BPMN	: Business Process Model and Notation
BPMod	: Business Process Modeling
CFP	: COSMIC Function Point
CMMI	: Capability Maturity Model Integration
COBIT	: Control Objectives for Information Technology
COSMIC	: The Common Software Measurement International Consortium
CRUDL	: Create, Read, Update, Delete, List
DG	: Data Group
DM	: Data Movement
EPC	: Event Driven Process Chain Diagram
ERD	: Entity Relationship Diagram
FAD	: Function Allocation Diagram
FP	: Function Point
FSE	: Functional Size Estimation

FSM	: Functional Size Measurement
FT	: Function Tree
FTD	: Function Tree Diagram
FUR	: Functional User Requirement
GQM	: Goal Question Metric
IDEF	: Integrated Definition for Functional Modeling
IIS-ERO	: METU-IIS Establishment of Research Opportunities Module
ITIL	: IT Infrastructure Library
KPI	: Key Performance Indicator
METU	: Middle East Technical University
METU-IIS	: METU Integrated Information System
OC	: Organization Chart
OCD	: Organization Chart Diagram
OOI	: Object of Interest
PAIS	: Process Aware Information Systems
PublicInvest	: Public Investment Project
RAD	: Role Activity Diagram
SADT	: Structured Analysis and Design Technique
SDLC	: Software Development Life Cycle
UML	: Unified Modeling Language
UPROM	: Unified Business Process Modeling
VC	: Value Chain
VCD	: Value Chain Diagram
WS-BPEL	: Web Services Business Process Execution Language

CHAPTER 1

INTRODUCTION

Business processes are among an organization's most valuable assets as they are reflections of organizational culture [1]. Business process modeling (BPMod) is a commonly used method to analyze, understand and define business processes. By means of business process analysis, processes of an organization either for the as-is or the to-be situation is revealed. This information is then transferred to a structured form of business process models. BPMod has become a common tool for various aims in organizations, either for "pure organizational purposes" like business process reengineering and process improvement, but also for other perspectives like workflow specification, project management, human resource planning, knowledge management and certification [2]. The purpose of BPMod can be grouped into three main headings; process definition, process analysis and process automation [3].

BPMod studies for process analysis and process automation have diversified due to the different purpose, audience and perspective of those models [4] resulting in disconnected modeling activities and diversified models. The problem is relevant also in other areas. Information captured in business process models is usually not utilized in a systematic way in other practices in the organizations. In our studies, we identified that business process model knowledge can be utilized systematically for the practices of requirements analysis, software size estimation, process metrics identification and process documentation. In this research, we propose a unified BPMod methodology, UPROM, to conduct business process and other analysis activities in an integrated way to develop a set of models. When models are developed by applying UPROM, a set of artifacts related to these practices can be generated. These artifacts are user requirements document and functional size estimation of the software for process automation and process documentation including process definition document, business glossary, process improvement list and process key performance indicator (KPI) list.

This chapter starts with a discussion on the background of the problem. Then, the problem is stated and the purpose and significance of the study are described. In the following section research strategy and research questions are presented. The chapter ends with the description of the organization of the thesis.

1.1. Background of the Problem

Increased process awareness and the emergence of business process management (BPM) discipline led to initiation of many process automation software projects in the last decade [5]. This kind of software can be generally named as process aware information system (PAIS) [5]. BPM is a holistic process management approach integrating business and technological aspects of an organization to achieve continuous improvement. BPM life cycle covers the phases of business process (BP) analysis and design, process automation, process enactment and diagnosis [6]. BPM tool suites have become popular, promising rapid process automation based on process models and speeding up even further by integration with predefined process definitions of ERP systems or similar.

Business process automation covers analysis, implementation and monitoring of processes. This automation can be achieved either within BPM life cycle utilizing BPM tool suites or in a more conventional approach. In any case, it is critical to analyze, reveal and capture the knowledge of business processes in the business domain and carry this information to the technological domain through software development phases in a complete and consistent way. The models defined to analyze, understand, describe and communicate the processes in the business side are called descriptive process models. On the other hand executable systems require detailed formal models to automate the operational processes [7]. The two business process models, descriptive business process models and formal models for operational processes, have distinct characteristics in terms of semantics, notation and modeling style; thus different process models are developed for them [8]. This study is focused on analysis and modeling of descriptive business process models. When we use the concepts of BPMod and business process model, if otherwise is not explicitly stated, we mean descriptive models describing the processes from the business perspective.

BPMod is one of the most significant means for transferring process experience into structured process knowledge. Process modeling methodologies are useful in defining existing processes for a better understanding and analysis to discover current problems as well as depicting the to-be processes. Business process modeling covers analyzing, defining and improving business processes of organizations in alignment with the organization's strategic objectives. BPMod has become a common tool in various areas like business process analysis, process documentation, process improvement and process automation. Process improvement is known to be one of the main objectives of the BPMod activities in business side [9]. There are many other disciplines that require the knowledge captured in business processes. Specifically, if an organization plans to automate its business processes, the knowledge of business processes is necessary throughout the software development life cycle (SDLC); mainly for the activities of requirements engineering and project management by means of software size measurement. Designing process performance metrics is also critical to ensure process improvement and to identify the measures to be automated in the system.

In all of these areas, the knowledge that resides in an organization to conduct business processes is critical to achieve diverse goals. This knowledge may be explicit, defined as process documents or models; or it may be inherent in the culture of the organization, where it designates the operations as organizational instinct. If an organization conducts BPMod activities, it creates opportunities to improve its processes while it captures valuable and high amount of knowledge that can be utilized for other purposes. If the organization aims to conduct activities related to other purposes, with the help of methodologies to transfer business process model knowledge to other activities; the related artifacts can be developed in a more complete, consistent and maintainable way. Studies to guide modelers for activities focusing on their goals are not easy to find in the literature. The relation between business process models and software, and utilization of process models in software development have been explored in a limited number of studies.

One of the areas that can utilize business process model knowledge in SDLC is requirements engineering. Business processes are essential for requirements analysis. Basically, business process models themselves are already defined as a means for "*for documenting, analyzing, improving, and ultimately codifying a set of business process requirements*" [10]. Business process models are used by business analysts and software engineers in many ways to gather requirements in software development [11]–[14]. Descriptive business process models act as the basic input for the technological BPM side; especially when the processes are to be automated using BPM tool suites [15, Pt. 3]. But usually, they are not sufficient to define user requirements in a structured and traceable way to explain how the automated system is supposed to behave to conduct each activity in business processes [16]. Organizations require a set of numbered and traceable requirements statements for scope management during project management. Literature study by Nicolas and Toval concludes that although there is value in generating textual requirements from models, approaches do not exist to form requirements sentences from business process models [16]. In software development projects, much of the effort spent on BPMod is duplicated for requirements analysis activities, while additional effort is needed for keeping models and requirements synchronized. The studies relating requirements analysis and business process models are few, but promising and open for enhancements [17]–[19].

Another area to benefit from business process models is software size estimation. The information on the software size is needed as early as possible in software projects so that reliable effort estimation and planning can be conducted in early phases of SDLC. Conventional early functional size estimation methods generate size at an early phase but result in subjectivity and unrepeatability due to manual calculation [20], [21]. Business process models are valuable for early size estimation, as these models are available before the requirements specification phase of a software development project, and provide systematic and objective data on the system. There are only a few studies that utilize business process models for size estimation [22], [23].

Development of process documents is another area that business process models can be utilized. Process definition document, sometimes called user manual, is an important artifact to describe processes in detail and communicate process knowledge to different users. Stakeholders need process definition documents to examine the processes from a different perspective, to add detailed information to the processes or both. Standards and reference models point out the need for existence of process definition documents and even define expected coverage. [24]–[26]. The development of process definition documents brings a burden to developers, as it is hard to develop, communicate and maintain a written documentation. Business process models already cover much of the information in process definition documents.

One of the aims of process modeling is to ensure consistent use of terminology throughout the organization. Business glossary covering domain terminology is used for this purpose [27]. Achieving an agreed set of definitions is difficult, especially when processes are defined in a decentralized way [1]. Business process models cover important information on the terminology, but this information needs to be consistently identified.

Business process improvement (BPI) is one of the purposes for BPMod [3]. BPI can be conducted by modeling as-is processes, and then switching to a separate set of to-be processes. But usually organizations analyze their as-is processes and start to design to-be processes on the same set of models. During this activity, organizations usually need a mechanism to state the improvements they identified for the business processes and save them for reporting. In this way, they aim to make the improvements visible. More complex solutions exist, as an example a tool enables modeling as-is and to-be processes as two separate but related diagrams, then compare the elements they include [28]. But for more straight and detailed reporting, there is a need to provide a simple storage mechanism for improvements and obtain them as a process improvement list report.

Process documentation like process definition document, business glossary and process improvement list may be developed in a standardized format and with less effort if business process models are utilized. There are a few studies and tools preparing process definition documents by using process models [29], [30]. There are further open topics to develop process documentation as requested by different standards and to fulfill different user needs.

The aim of defining metrics for process performance is to guide the organizations to operate compatible with their organizational goals. A vision is founded in this area in the public sector in Turkey with the utilization of performance-focused budgeting [31]. Monitoring the execution of business processes is as important as automation of them. Process metrics, which are the metrics that require business data [32], focus on evaluation of process performance rather than the systems. Process metrics are defined on business process models at business level analysis. This definition is hardly utilized in the technological side as it does not specify how the related data is collected.

Obviously, business process models carry important information for defining process metrics. In a study we conducted, we utilized process models to develop an easily collectible, process-related, goal oriented metric set [33]. We got lessons learned from this study that can be used to establish process metrics set related to process models.

In summary, we observe that the structural transfer of business process knowledge to related artifacts is critical to prevent reworks and improve the related artifacts. There are a lot of practices we have identified for which business process models can be utilized to develop the artifacts in a complete, consistent and maintainable way.

1.2. Statement of the Problem

We observe that information captured in business process models is usually not utilized in a systematic way to develop artifacts that can benefit from this information. "*A wide variety of BPMod techniques have been developed over the years based on different theoretical foundations and for different purposes*" [34]. If an organization has more than one purpose, frequently, different documents and models are developed and information captured due to one is hardly utilized for others. This increases the development effort of those artifacts, makes completeness and consistency hard to achieve, while also results in broken traceability between them and business processes. Later, as business process definitions and other artifacts are updated separately, they may become unrelated and conflicting; and much effort is spent for maintenance. The problem is even more critical if the organization plans to automate those processes by business application software, or synonymously PAIS.

Business process models are usually developed earlier than other artifacts, awaiting ready for usage. Current methodologies do not create business process models transferable to other activities. Even if an organization is sufficiently aware to communicate existing studies among teams, methodologies do not guide users to systematically utilize models for purposes other than what they were originally constructed for. To prevent this problem, integrated methods have to be applied to software engineering and other organizational activities so that related artifacts utilize business process information in the most structured way.

Utilizing literature review results and industrial experiences, we identified that there are opportunities in the practices of requirements analysis, software size estimation, process metrics definition and process documentation to utilize business process models. If an organization plans to achieve these goals, it can utilize its business process models to develop the related artifacts. In this study, we aim to find a solution to utilize business process models to systematically develop artifacts for these practices. Our solution includes a BPMod methodology, including modeling process, notation, metamodel, rules and guidelines. We intend to develop a set of models following this methodology so that the artifacts can be generated automatically.

To sum up, we emphasize that if an organization develops business process models, by systematically capturing the knowledge in business processes, that organization creates an opportunity to transfer this knowledge to other areas; for which it can create related artifacts in a complete, consistent and maintainable way.

1.3. Purpose of the Study

The purpose of this study is to develop a unified BPMod methodology that provides an integrated approach supporting BPMod targeted for diverse goals. Such a BPMod methodology shall enable users to transfer this knowledge to artifacts to be developed for other goals. As a result, these artifacts will be complete, consistent, maintainable and traceable to business process models.

The diverse areas we reveal that can benefit from BPMod activities are user requirements analysis, software size estimation, process documentation and process metrics definition. It is aimed to include process, notation, metamodel, rules and guidelines in the methodology to provide analysts a full approach to analyze and develop process models while achieving artifacts for these diverse goals in an integrated way. Main activities to be followed to apply the methodology are analysis of descriptive business processes, analysis of functions allocated for process automation and then generation of related artifacts by using the completed models. The high level process can be seen in Figure 1.

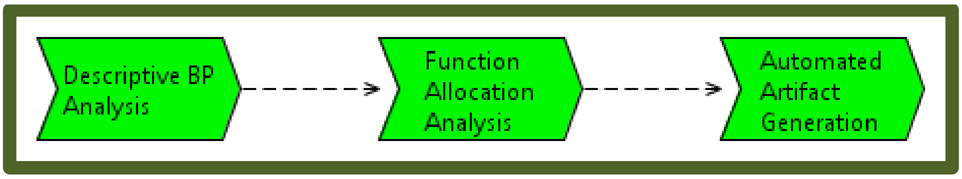


Figure 1 High level UPROM process

A methodology is "*a documented approach for performing activities in a coherent, consistent, accountable, and repeatable manner*" [35]. Modeling methodology, modeling language and modeling tool is found to be the most critical success factors amongst modeling related factors [36]. The notation of UPROM methodology constituting the language includes diagram types, relations, semantic and syntactic rules for the notation. We aim to use BPMod notations widely accepted in practice and literature where applicable, extending them with extra elements and constraints. The notation is formalized with a metamodel. We aim to achieve a modular and adaptive notation that the user can add or remove features according to her goals as guided by the methodology. A BPMod process, as part of the methodology, describes the application of the methodology. Rules and guidelines as part of UPROM aim to guide users on how to apply the methodology and use the notation to achieve diverse goals required by the user. The components of UPROM methodology can be seen in Figure 2.

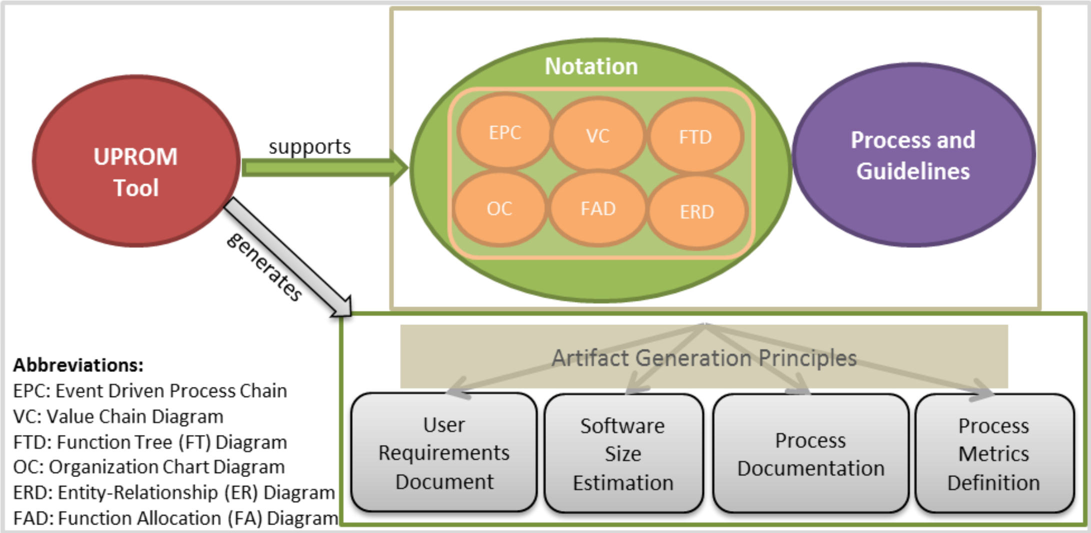


Figure 2 UPROM methodology components

1.4. Significance of the Study

Integrating business process modeling activities for diverse goals can provide many benefits in practice, while it can associate different studies in literature. Modeling methodologies improve learning curve, increase reliability of the models and shorten the required effort [37], [38]. Lack of guidelines for modeling activities result in ad hoc approaches, and most of the time a natural language based description of the process is formed. Resultant process descriptions are error prone and difficult for structured analyses. In this sense, developing business process models with a standardized notation enhances interoperability both within an organization and between organizations. In this way the models developed using this approach may be utilized in extensive studies like e-government projects. Basically, defining a methodology including the process and guidelines already brings benefits of graphical BPMod approaches like enhancing efficiency of processes, revealing inconsistencies, disseminating standard definitions of models, reducing paperwork and bureaucracy, conformance to standards and regulations. With the inflated number of BPMod techniques, there is a lack of comprehensive “BPMod Methodology” description in the field [34], [39, p. 28]. Keeping this in mind, we provide a methodology including notation, process and guidelines.

Guiding different groups of experts with the same methodology allows synergy to be created in terms of modeling studies. Rather than modeling separately for diverse goals, utilizing a unique notation for multiple goals provides synergy for experts in different areas. In the long run, utilizing a common language and notation for process models also enhances interoperability between organizations having interfaces to each other.

Analyzing and defining business processes is especially critical when the processes are complex and many people are involved in the processes [34], [40]. With the increased process awareness, many process automation projects are initiated in the last decade [11]. BPM tool suites, workflow management systems built upon high level process automation infrastructures, or any other software system developed by lower level or general purpose coding languages are used to develop the software for process automation. For any of these solutions, analysis of business processes from the user perspective and identification of requirements to automate those processes still remain as a major software development activity. The quality of requirements is critical to lower project costs, as the requirements defects are the most costly defects to correct in SDLC [41]. Business processes are an important input for requirements, but they are usually utilized only as abstract and conceptual inputs by software designers to model the operational processes or to design the automation software.

UPROM ensures that business process knowledge is transferred to requirements analysis phase by analyzing the functions to be automated based on business process models. In this way, the repetition of effort to gather the process knowledge is prevented, descriptive and formal operational processes remain aligned, traceability between the business and technological artifacts are kept and artifacts are easily maintained. A unified approach also brings other benefits like providing a better communication environment between customers and developers, ensuring that process owners and software engineers are on the same terms, allowing process knowledge to be used within the requirements phase [42], revealing relations between models and requirements, exposing information systems integration points within business process models and in these ways, improving completeness and traceability of requirements [16].

Functional size of software is a base measure which can be used for various software management tasks such as planning, monitoring and control [43]. Early size estimation is critical for successful project planning. As early functional size estimation methods do not provide a clear measurement manual, the reliability of the methods directly depends on the estimator. This results in subjective application of the methods. Estimating the functional size by using business process models is a new approach and a few studies are found with a similar approach [22], [23]. In UPROM, measurement rules and procedures are standardized and even automated, assuring that the measurement is independent from the measurer. Additionally, by automating measurement, the effort is reduced significantly, estimation repeatability is provided, no expertise in measurement is required and estimation is easily maintained when updates are introduced to the models.

In the context of this study, by "process documentation", we mean to cover process definition document, business glossary and process improvement list. Process definition documents or user manuals are stated as a requirement by many organizational standards, if not already needed by the organization itself to express the operations to be conducted in a detailed and standardized way for different user types. In public organizations, these manuals expose themselves as regulatory documents; while in the private sector they are mainly process and procedure descriptions. Some of the organizations require process documentation together with process models. The reason for that is usually twofold: Firstly, they usually want to capture detailed information that is hard or almost impossible to depict in the descriptive models; or that decreases the readability of the models as they increase the complexity of the models unnecessarily. The second reason is that, due to the background and expertise, some of the employees in an organization prefer to read process documentation rather than reading models. Development of user manuals takes considerable amount of time within the SDLC. It is hard to prepare them in detail and in a standard way. Additionally, organizations usually develop a business glossary to assure that all readers of the process documentation have a common understanding of the terminology. But the glossary usually becomes out of date, as it is not updated synchronously with the process definitions, or conflicting descriptions are placed. Also, the management usually requires the improvements obtained by business process analysis studies to be visible, but it is hard to identify and keep track of a list in parallel to modeling activities. An environment to record those improvements is observed to be useful in many BPMod initiatives. Thus, utilizing business process models for generating process documentation provide significant benefits, as the documents will be prepared in a standardized way, the updates will be conducted on the single source of models, the documents will have high maintainability as they will be regenerated in case of any changes in process descriptions.

Establishing a process metrics system based on process models enables process focused, adaptive and measurable performance infrastructure. A lot of performance management infrastructure binds the performance management to the organizational processes [44]–[47]. In this way performance system is ensured to be based on processes and strategic goals. When an organization requires integrating performance management system with software infrastructure, the points of data collection and data analysis requirements needs to be clear. As process metrics are determined related to process models, the changes in processes will be easily reflected to performance management systems. When the use of performance management and business process modeling concepts are analyzed in the literature, it is observed that the concept of performance management is basically used in process workflow systems; as defined metrics to be measured are collected during the execution of workflow activities. But there is not many studies focusing on designing metrics on the business process model infrastructure of an organization. UPROM aims to provide a process metrics definition method that guides users to define how the metrics will be collected and analyzed during process modeling. The resulting definition of process metrics can easily be used in the development of the process automation software.

To sum up, during software development, separate efforts are spent in activities like requirements analysis, software size estimation, development of process documentation and definition of process metrics. The unified BPMod methodology proposed in this study aims to create a significant opportunity to integrate these activities and in this way increase the completeness, consistency and maintainability of the related artifacts and enhance their traceability to business processes and maintainability.

1.5. Research Strategy

The objective of this research is to develop a unified BPMod methodology to integrate activities and achieve goals in diverse practices. The idea of utilizing business process models for diverse goals was initiated by means of our experiences in the BPMod field. We started the research activities with a review of the literature. The literature review is conducted on business process modeling and focused especially on the usage of business process models for different practices, especially in requirements analysis, software size estimation and process documentation. The literature search is conducted based on keywords on literature search engines and libraries.

Based on the literature review results and previous experiences, we identified improvement opportunities in the field and formulated our research questions. We conducted narrower studies to identify how we can utilize business process models for each practice and developed the methodology for each. We then aggregated the methodology to cover all practices in an integrated way.

To verify and validate the applicability of the methodology and answer research questions, we designed and applied multiple case study research for four cases. Each case aimed to evaluate the methodology from different perspectives. We collected metrics, evaluated direct and participant observations, and conducted semi-structured interviews as multiple sources of evidences. In the last step, we gathered the results from the case studies, answered the research questions and provided the discussions.

1.6. Organization of the Thesis

This chapter includes a discussion on the background of the problem, and problems observed that triggers the study planned. The purpose of the study is presented before the discussion on the significance of the study, covering how the results of this study will have an effect on the solution of the problems. Next, research questions are stated.

Chapter two is a review of the literature in business process modeling and the usage of business process models in the related areas of our methodology.

Chapter three presents the methodology we propose in this study in detail. The notation and metamodel, process and guidelines for BPMod and artifact generation procedures are described as part of the methodology. UPROM tool, the prototype tool developed to apply UPROM is also presented.

Chapter four presents the implementation of case studies and references the outputs of these studies. Findings and lessons learned are described and results are discussed.

Chapter five describes the overall findings, achievements and planned future work.

CHAPTER 2

RELATED RESEARCH

The purpose of this chapter is to review the research literature to identify the BPMod notations currently used in practice and examined in the literature. It is important for the purpose of the study to understand the current BPMod notations to get insight on how BPMod is used in the literature, different characteristics of notations and their existing usage. By understanding existing approaches, it is possible to judge how these approaches can be utilized and extended to develop a unified methodology for diverse goals. This chapter also includes review of research on different uses of BPMod notations. Special attention is paid to comparative studies as they provide more insight on multiple approaches and possible uses for them. Section 2.1 provides the definitions of the related concepts in the field, and highlights the relation of business process models with those concepts. In section 2.2 related work on BPMod notations are summarized. In section 2.3 our rationale for selecting BPMod notation is explained. Section 2.4 briefly explains existing BPMod methodologies and tools used for BP analysis. Finally in section 2.5 the studies focusing on usage of business process models for diverse goals are examined in subsections.

2.1. Business Process Management and Business Process Modeling

Before delving into details of BPMod literature, we start from the definition of business process and then move to a brief definition of business process management (BPM), as it is a wider area of study which covers BPMod as a major activity.

In its simplest definition provided by Davenport [48, p. 5], "*a process is a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action*". Moreover, Becker et.al. [49, p. 4] clarifies the definition of business process, indicating that "*a business process is a special process that is directed by the business objectives of a company and by the business environment*". Here, instead of "company", we will rather use organization, as business processes and BPMod are applicable to different organization types. Scheer and Nüttgens [50, p. 1] shortly define business processes as "*a procedure relevant for adding value to an organization*". As Gartner states, "*processes span organizational boundaries, linking together people,* *information flows, systems and other assets to create and deliver value to customers and constituents*" [51].

Business processes are in the center for many disciplines and approaches like BPM, BPI, business process reengineering (BPR), enterprise modeling, workflow management and process aware information systems (PAIS). Process-aware information systems support operational processes, or "workflow processes" [5]. It is a general term used to indicate systems that, in some way, manage and execute workflows [11] as part of BPM systems or in other ways. Here, we especially focus on BPM, as it is an extensive and holistic approach covering all activities related to business processes. Weske emphasizes that "*BPM includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes*" [6]. The activities conducted on business processes during BPM activities are "*design, administration, configuration, enactment and analysis*" [6]. It not only includes the activities related to managing business processes, but also methods, techniques and tools [52]. Gartner also defines BPM as "*the discipline of managing processes as the means for improving business performance outcomes and operational agility*" [51].

BPM as a holistic approach covers six core factors; strategic alignment, governance, methods, information systems, people and culture [15]. BPM includes all activities of a business process lifecycle from design to configuration, enactment and evaluation [6].

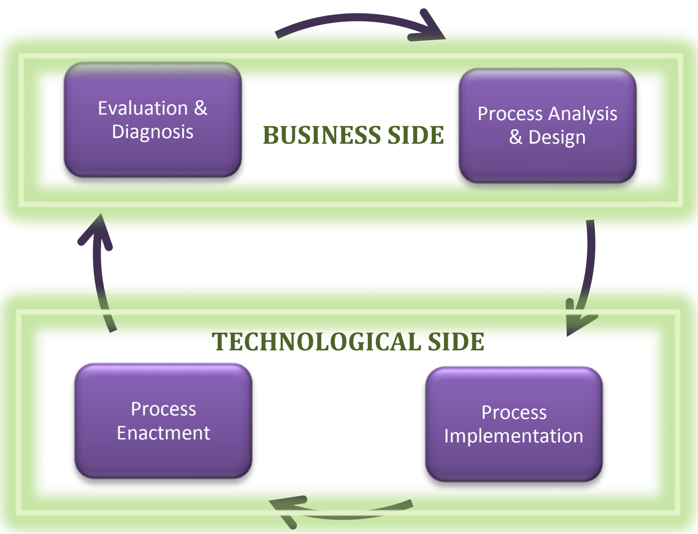


Figure 3 BPM lifecycle

Business process management approaches are handled as a lifecycle, similar to software development. There are life cycle models offered by different authors but similar to each other like [5], [53]. Here, we prefer to describe the approach of Muehlen, as it integrates other approaches and clarifies the place of BPMod in BPM. Muehlen specifies four basic steps for BPM shown in Figure 3. BPM starts with analysis of the environment and the organization. Then, as the first step, process analysis and design comes, which includes the activity of BPMod and develops business process models as the output. Next step, process implementation includes forming the infrastructure to execute processes in the way determined by business process models. If automation is required, information systems are developed here. In the following step, process enactment, the process instances are executed using the infrastructures. During this stage, process monitoring is conducted and process metrics are collected. In the process evaluation phase, inputs from other steps are utilized to evaluate the processes. The outcomes of this step retriggers process design step. We see that business process model is the key outcome of BPM life cycle, and have a significant effect on the conduct of this life cycle.

Modeling can be defined as abstraction of reality for the concerned domain. A process model "*is an abstract description of an actual or proposed process that represents selected process elements that are considered important to the purpose of the model*" [37]. According to the general model theory, there are three basic features of models [54]:

- Mapping feature indicates that models are always associated with something, which may exist in real life, or may be imaginational things or models.
- Reduction or abstraction feature explains that models contain less information compared to the original thing the model is prepared for. The reduced attributes of the model with respect to the original ones are not necessary for the goal to use the model.
- Pragmatism feature expresses that models are not created in vain, they are used to represent the originals and appropriate to be used in substitute of the originals.

Within the scope of this study, what we mean by business process models are graphical representations of processes. However, there are other alternatives for models, like natural language or code. BPMod in this sense is the activity of developing business process models with some specific goal(s). The main building block of business processes are activities. To achieve specific business goals, activities are performed "in coordination" [6].

The ideas and concerns of process modeling considered today dates back to nineties. Curtis and Kellner [37] described concepts of business process reengineering, workflow systems, and distributed automated system requirements. Software process modeling and business process modeling definitions are made within this study. Perspectives of process representation are defined as functional, behavioral, organizational and informational, and the importance of expressing different perspectives in the models is emphasized. A basic issue of process modeling is defined as the level of formality required by different aims: execution and informatory. This question is still in place.

Together with its role in BPM, business process models have many usages as mentioned in Chapter 1. Some typical purposes of BPMod can be listed as documentation, process-oriented reorganization, continuous process improvement, benchmarking, knowledge management, selection of ERP systems, human resource planning [2], certification like ISO 9001, Sarbanes-Oxley Act, activity-based costing. For every different purpose, different approaches and notation evolved in the literature in the last 20 years [23]. Within the scope of our study, we first provide a brief overview of common BPMod notations in the literature that we can use in our methodology. Then, a summary of existing BPMod methodologies is provided. We can then move forward to studies focusing on usage of BPMod for different goals.

2.2. Business Process Modeling Notations

Although BPMod basically aims to resolve the complexity introduced by the system and related goals, the BPMod methodologies themselves bring complexity and cost [49]. For the various purposes emerging, various BPMod approaches are developed. The approaches range from simple techniques like flow charts, UML activity diagrams [55] to more complex and even formalized ones like event-driven process chains (EPC) [56], Petri nets [57], BPMN [58], WS-BPEL [59].

Business process modeling techniques in terms of formalization can be divided into two. Descriptive business models utilize semi-formal graphical modeling techniques. They are focused on the analysis of business processes from the business perspective; establishing a communication environment to understand and improve processes with domain experts [4], [34]. These models conform to semantic and syntactic rules, but they do not need to conform to strict formalization like the ones required in execution models [39]. Examples of these are EPC, UML activity diagram, role activity diagram (RAD) [3], IDEF family [60] and depending on the purpose and usage, BPMN. IDEF3 Process Description Capture Method [61] is specifically developed for BPMod and has a scenario-driven process flow description approach. It captures both descriptions of process flow and object state transitions [62], and have a similar approach with EPC in terms of this functionality. On the contrary, formal modeling techniques are "*founded on mathematical, rigorous paradigms*" [34]. They are mostly used for process automation purposes [7]. Other typical usages include formal process analysis [57], simulation and experimentation [63].

Another categorization of business process model approaches is made according to business process dynamics, that is, the flow [64]. Business process models established on input-output flow focuses on passive participants manipulated by activities. An example of such an approach is IDEF0 [60]. In the workflow, the focus is on the timely flow of activities; which is the most common notation with many examples including IDEF3 [61], UML Activity and Petri-nets. In agent-related view, the model is organized for the agents conducting the work. Role-Activity Diagrams (RAD) is an example of this view [3]. In the state flow category, the focus is on changes produced by the process; like state-transition diagrams.

Petri-Net is a formalized mathematical modeling technique which is rather utilized for more structured purposes like process analysis, process execution [65]. Formalized Petri-Net model includes static and dynamic properties of a system that results from its execution [66]. It is a very old technique going back to the sixties, and many extensions have been introduced to enhance it. YAWL is a new workflow language extending Petri-Nets to support all workflow patterns, providing an open source framework for modeling and also supporting execution [67]. Other widely known process modeling techniques (which are also utilized for many other purposes in the literature) are Data Flow Diagrams (DFD) [68] and ANSI flowcharts [69].

There are other BPMod studies having different perspectives. Caetano et.al. [70] proposes a BPMod approach defining business objects and roles as separate modeling activities. To associate objects and roles, intrinsic and extrinsic features of objects are utilized. By means of role-based BPMod, the models become more understandable and patterns between business objects and roles become explicit. Studies on distributed modeling and modeling based on different subjects like subject oriented BPMod (S-BPM) also support role based modeling [1], [71]. It is also possible to group BPMod approaches as graph-based and rule-based, as done by [72]. It is possible to observe how these characteristic has affected the properties of expressiveness, flexibility, adaptability, dynamism and complexity for the approaches. Some studies introduce extensions to existing approaches to cover some deficiencies observed [73]. Commonly used BPMod notations in literature and practice are described in more detail in subsections below.

2.2.1. Event Driven Process Chains (EPC)

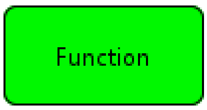
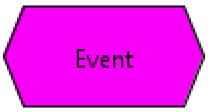
Event Driven Process Chains (EPC) is one of the most common descriptive BPMod techniques encountered in many of modeling languages [74], [75]. An EPC is an "*ordered graph of events and functions; it provides various connectors that allow alternative and parallel execution of processes*" [76]. EPC notation is introduced by Scheer's studies and ARIS methodology [56]. EPC is used to represent control flow and behavioral view of business processes.

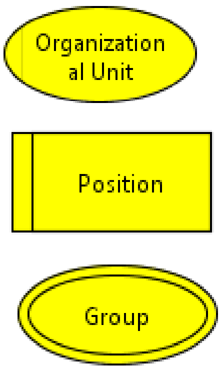
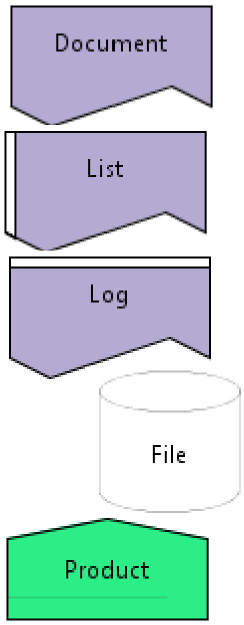
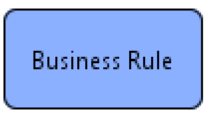
In its simplest version, EPC notation includes constructs of function, event and logical operators, as described in Table 1. With the control flow connection between these constructs, the occurrence of functions and events is depicted depending on each other and in order of time. An EPC diagram always starts and ends with events. Regularly, functions and events should be alternating. But in practice, trivial events are usually omitted. A hierarchical structure can be established by detailing a function in another process which is assigned as a sub-diagram of that function. Logical operators are used to model splits, joins, alternative flows and loops. Some examples of usage for these operators are shown in Figure 4.

Basic EPCs, which contain only function, event and operator constructs are enriched to contain behavioral aspects of the processes, called as extended EPC (eEPC). In this study, we use the abbreviations eEPC and EPC interchangeably, as most of the applications include extended construct set of the notation. The constructs commonly used in the extended notation are shown in Table 1. Basically, organizational elements are utilized to model the organizational view. They represent the responsibility to conduct a function by being connected to functions. Information carrier constructs are utilized to show the informational view, being connected to functions as inputs and outputs. The constraints for connections between the constructs of the notation and connection types are shown in Figure 5.

EPC is the central notation of ARIS framework integrating functional, behavioral, organizational and informational views [56]. An example process modeled with EPC notation can be seen in Figure 6. There are studies to define formal semantics of EPCs and conduct formal analysis based on them [77], [78]. Another study showed that with small updates, EPC notation can support most of the workflow patterns [79]. These studies exhibit that formal analysis can be conducted on EPCs if the diagrams are developed based on formal semantics. However, it is hard to assure conformance to formal semantics during process analysis in the business domain. Semantics of EPCs explain the flow behavior of the diagrams considering the splits used in EPCs [77], [80], mainly expressed by soundness of a process flow [65]. Researchers tried to formulate the informal semantics of EPCs by formal semantics. This is basically assuming tokens flowing through the arcs of an EPC diagram and examining the behavior of those tokens and checking the soundness of the overall diagram. Due to the nature of splits and joins, local and non-local semantics of EPCs differ from each other; which cause some flaws in the definition of EPC semantics. The semantic definition of EPCs is critical for automated workflow systems, not the business process analysis activities. However, the flaws in the semantics point also to logic problems in the business process analysis activities; which brings the necessity to provide relevant guidance in our methodology, too.

Table 1: Construct set of eEPC Notation [81]

Construct symbol	Description	Example
	**Function** is a technical task or an activity performed on an object to support one or more business objectives.	Review a Document, Send a Message, Create a Record, Update a Report, Archive a Document.
	An **event** represents a state of an object that is relevant in terms of business context, which controls or influences the further procedure of the business process. Events trigger activities and are the results of	Document Reviewed, Message Sent, Customer Arrived, Message Received.

	activities. An activity is a time consuming occurrence while an event is related to one point in time.	
	**Logical Operators (And, Or, XOR)** represent alternations that link events and activities in process chains.	
	**Process Interface** provides modularity by referencing another process which is not a sub-process of the existing process.	Procurement, quality assurance
	**Organizational Elements:** **Organizational units** are the performers of the tasks required to attain the business objectives. The smallest organizational unit in a company is a **position**. It is assigned to employees (persons). A **group** may represent a group of employees (persons) which are working together for a specific period of time.	Design Dept., Finance Dept., Project Team, Review Team, Project Manager, Designer, Customer.
	**Information Carriers:** **Document** is a type of information carrier, which represents a means to store and transmit information. It can be in the form of a document, an email, a fax, a CD, or a verbal message that is produced out of an activity or input to be processed by an activity. **List** and **log** objects are other symbols representing data types. **File** is a storage for a set of data, while **product** represents the resultant output of a function.	Software Requirements Spec. (SRS), Review Form, List of…, Notification Email, Checklist.
	The rules which constraint how the function is executed.	The assignment must be submitted in two weeks

	An application type represents a system or a tool that is used to support agents (actors, roles) in performing their activities.	Req. Mang. Tool, Project Mang. Tool, or specifically MS. Project, Rational Rose, etc.

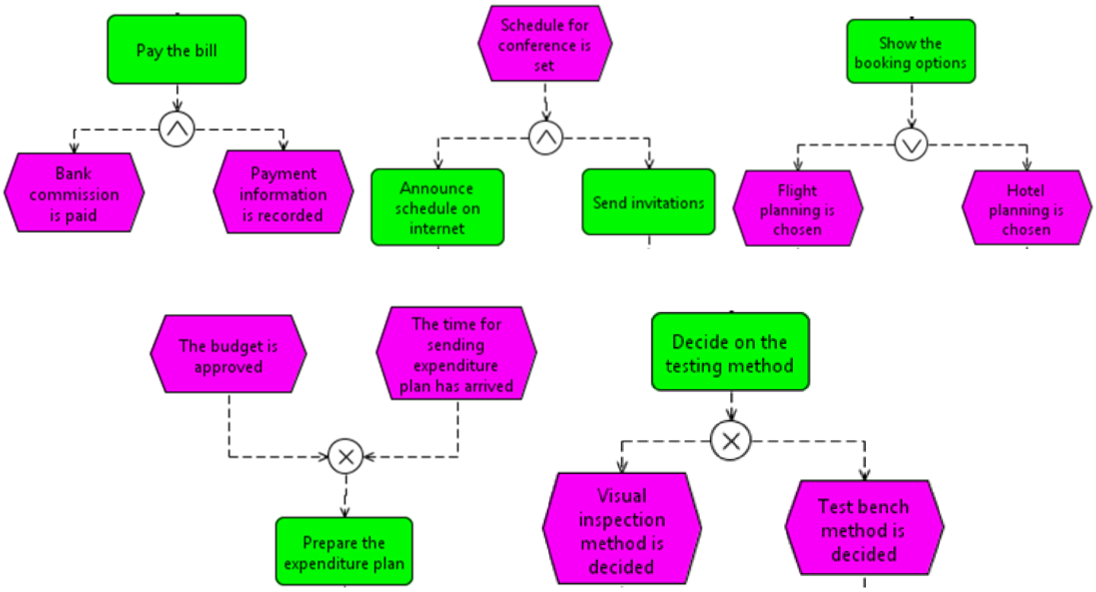


Figure 4 Usage of logical operators in EPC

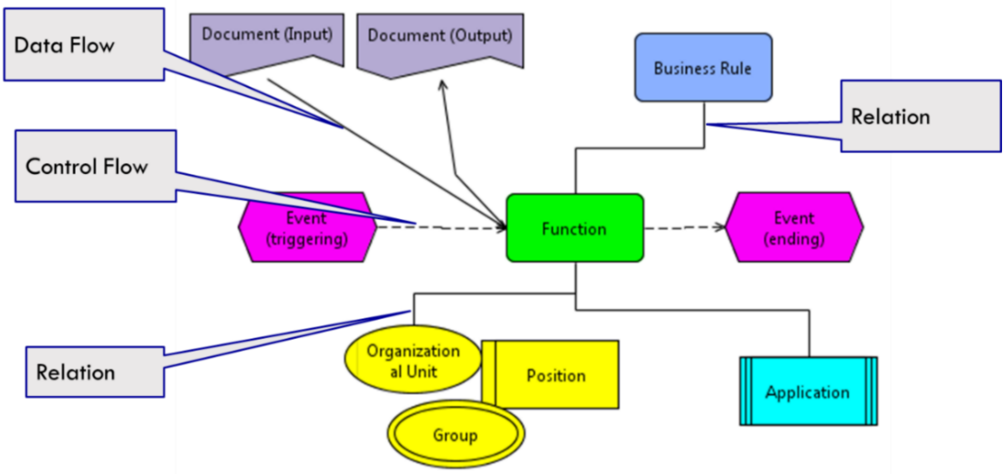


Figure 5 Connection constraints between the constructs of eEPC

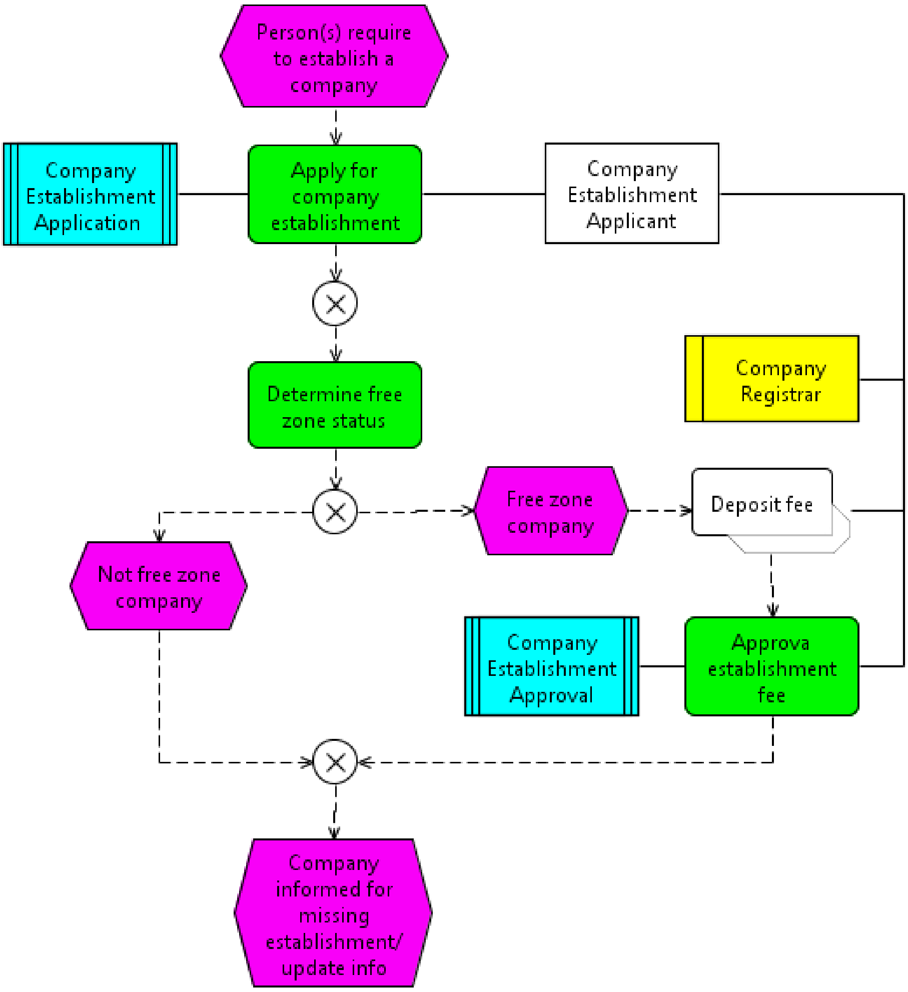


Figure 6 An example EPC diagram

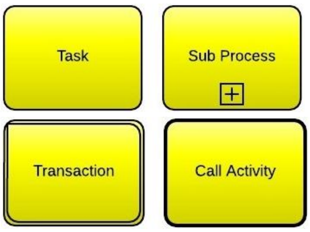
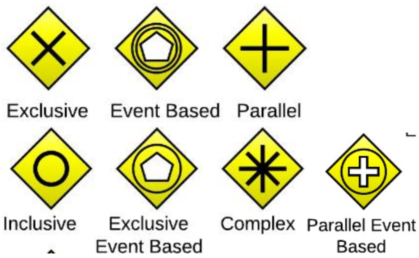
2.2.2. Business Process Model and Notation (BPMN)

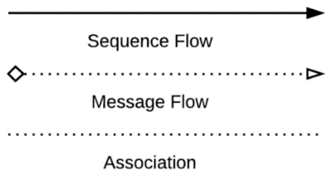
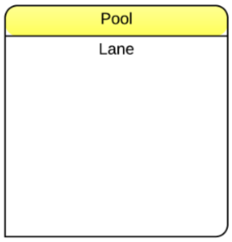
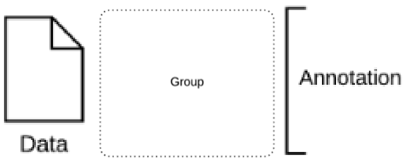
BPMN [58] is a standard developed by OMG that aims at disseminating the usage of BPMod to different types of users including business and technical users. BPMN is extended in the version 2.0 published as of March 2011. The standard with the extended notation focuses on using the same notation for both process analysis and process execution. Core BPMN notation is very similar to UML activity diagram, while it has many opportunities to make extensions on regular usage of activity flow notations [82]. It also supports a well-known business process execution language, BPEL (or WS-BPEL) [59] by means of a mapping between graphical notation elements of BPMN and language constructs of WS-BPEL. WS-BPEL is a modeling language that enables the development of executable business process models for web services.

BPMN method uses business process diagrams (BPD) to model business processes. Constructs of the notation can be grouped as flow objects, connecting objects, swim lanes and artifacts. The constructs are depicted in Table 2.

Due to the notation supporting execution, BPMN had many constructs for behavioral and functional perspectives. In its earlier versions, support for the organizational and informational perspectives was weak. Also, BPMN had problems for its meta-model. It didn't have a well-defined meta-model, thus making it hard to transform to other languages. BPMN 2.0 was developed to overcome these problems. An example BPMN model can be seen in Figure 7. With the enriched set of constructs, BPMN is good in expressiveness, on the other hand has high complexity [34]. A previous study showed that BPMN users used only 9 constructs typically in their models [83].

Table 2 Construct set of BPMN notation [58]

Construct symbol	Description
Flow Objects	
Events: 	An **event** denotes something that "happens". Three types exist as shown on the left. In the extended notation, events exist for **catching** (an incoming message) or **throwing** (a message) messages.
Activity: 	It describes a work to be done. It can be a single unit of work, **task**; a reference to a detailed process, **sub process**; a set of activities completed as a whole, **transaction**; and a reference to a global process**, call activity**.
Gateway: 	Gateways determine different types of forking and merging paths. **Exclusive**, **inclusive** and **parallel** types correspond to **and**, **or** and **xor** logical operations. **Event** based gateway types act on the occurrence of events.
Connecting Objects:	

	**Sequence flow** is used between flow objects to depict the control flow. **Message flow** is used between pools. **Association** is used to connect notes or texts.
Swim lanes:	
Pools and lanes: 	**Pools** separate different organizations. Pools accommodate one or more lanes inside. **Lanes** are used to organize activities according to roles.
Artifacts:	
Data objects, groups and annotations: 	**Data** objects represent any kind of data. **Group** is used to group data but doesn't have semantic meaning. **Annotation**s are used to add explanations.

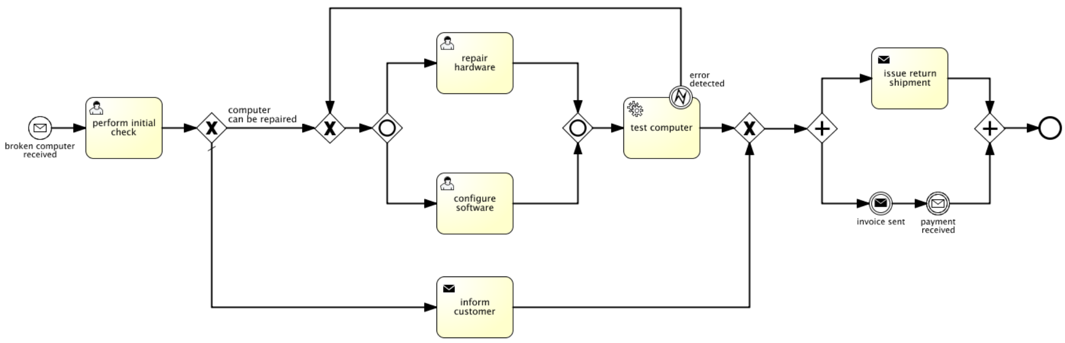


Figure 7 An example BPMN diagram [84]

2.2.3. Integrated Definition for Functional Modeling (IDEF)

IDEF is a family of modeling languages based on previous studies in systems engineering. IDEF language set has more than sixteen diagram types [85]. Five of them mostly used are listed below.

- IDEF0 – Function Modeling Method
- IDEF1 – Information Modeling Method

- IDEF1X – Data Modeling Method
- IDEF3 – Process Description Capture Method
- IDEF4 – Object-oriented Design Method

IDEF0 and IDEF3 are the notations utilized to analyze and model business processes [85]. IDEF0 is built upon the Structured Analysis and Design Technique (SADT) which is a functional modeling language. It is utilized to analyze individual functions in detail. It doesn't provide control flow modeling. Flow of functions is revealed by means of dependencies between the functions. It is used to represent data flow and functional flow of processes. The functional analysis on each function is conducted based on the notation shown in Figure 8.

IDEF3 was developed to meet the requirement to express control flow in the processes [61]. It describes the flow in a scenario driven style. It represents the behavioral perspective with process flow descriptions and object state transitions. In IDEF3, the term description is reserved for records achieved by observations or experience, where the model is used for idealized system of objects and properties.

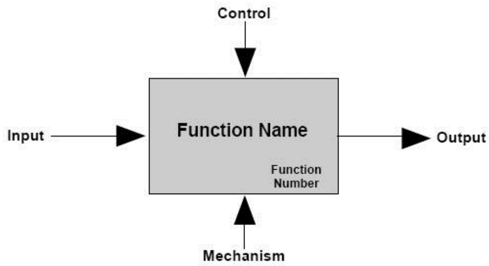


Figure 8 IDEF0 Notation

2.2.4. UML Activity Diagram

The Unified Modeling Language (UML) is a standardized modeling language by OMG utilized for many purposes, basically focusing on object-oriented software design and development [55]. UML activity diagram within the UML notation family is used for process modeling. The notation of UML activity diagram includes start and end states, activity, decision point, fork, join and control flow. Each diagram starts with an initial state and ends with one or more final state. Control flow is depicted by connecting elements with a control flow connection. Alternative paths are shown with decision points, and parallel execution in between forks and joins [86].

UML activity diagrams are not developed for the purpose of BPMod. However, duty its simplicity and availability in commonly used drawing tools, it is heavily utilized for this purpose [87]. However, it is weak in expressing organizational and informational views. An example UML diagram can be seen in Figure 9.

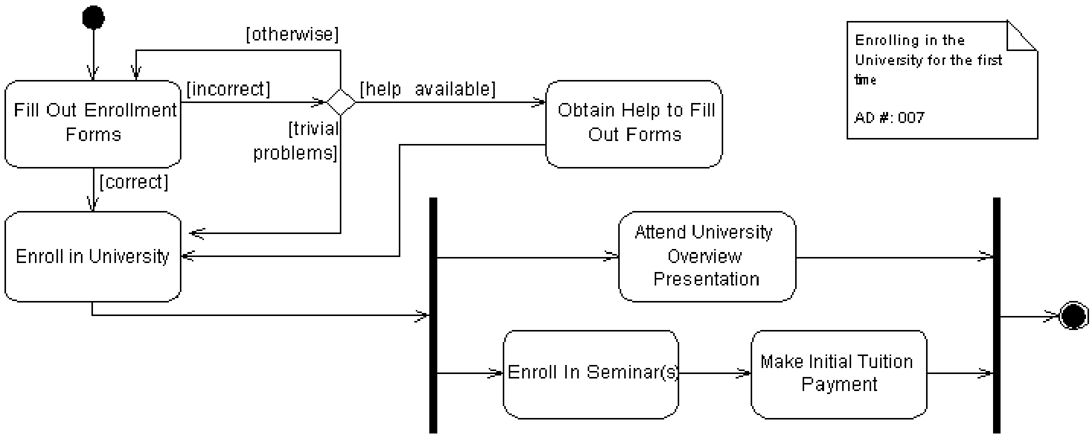


Figure 9 An example UML 2.0 activity diagram [86]

2.2.5. Role Activity Diagram (RAD)

RAD notation expresses the processes based on the roles. RAD is used for analysis, modeling and elicitation of business processes [3]. A process is depicted as a set of roles interacting with each other. A role can be a person, a group of people or a system. A role contains a set of activities. Other than that, decisions and transactions are used to model the flow of activities. Roles interact with each other. Figure 10 shows the notational elements of RAD. RAD is powerful to represent the organizational view and the interactions between the organizational elements and systems. On the contrary, it is weak to express informational perspective. Also, it is hard to express decomposition of processes. They don't have underlying formal semantics.

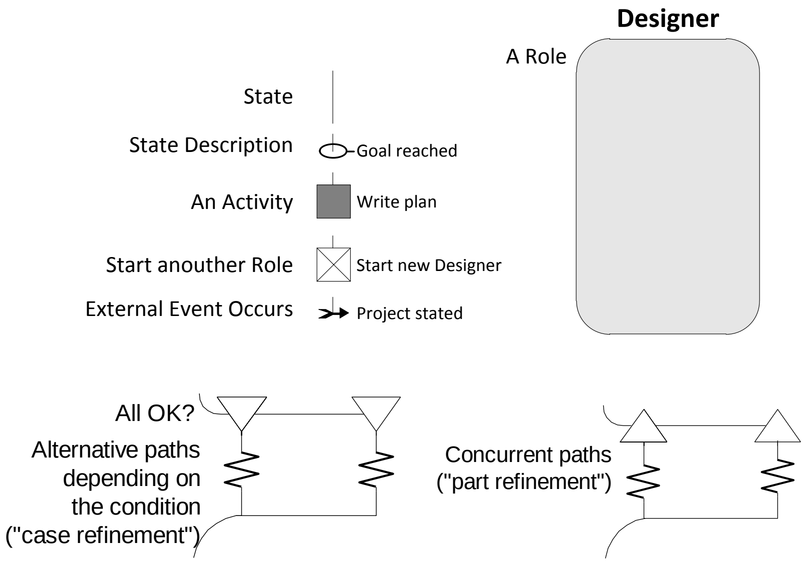


Figure 10 RAD notation

2.3. Selection of the Business Process Modeling Notation

There are studies to evaluate the expressive power of BPMod notations and provide a benchmark and comparison basis. To determine fundamental requirements of BPMod on a structural and standard way, Workflow Patterns Initiative is founded [88]. In their extensive report, the group identifies control flow, data and resource flow patterns [67]. Workflow patterns are utilized in many studies to evaluate the expressive power of BPMod languages. Bunge-Wand-Weber (BWW) model is used to evaluate representational capabilities regarding the modeling techniques as a reference benchmark [34], [89]. These researchers suggest a method for representational analysis of BPMod techniques based on ontologies. Lu and Sadiq [72] propose an approach to compare BPMod approaches by their expressiveness, flexibility, adaptability, dynamism and complexity. These approaches provide precious idea on the characteristics of BPMod techniques and their potential to be utilized in different ways. Obviously, it is not easy to cover all required dimensions of BPMod languages with such models, also we cannot expect a descriptive BPMod language to fully meet all the requirements to represent the domain fully. Still, in such studies, we see that amongst other descriptive BPMod notations, BPMN and EPC take place to have higher representational capabilities. Thus, we focus our analysis on BPMN and EPC.

EPC and BPMN are two notations on which a lot of "versus war" goes on between competitors [90]. They are popular notations, and they have advantages over each other. [91] indicates that BPMN is effective in covering control flow patterns, while not expressive in terms of resource patterns. On the other hand, EPC lacks to support many workflow patterns [79]. EPC is basically a notation on business side used to analyze business activities, whereas BPMN is a notation to compose processes in technology side, and then utilize them to execute processes. To bridge the business and technology aspects of processes, we need to keep this difference in mind.

Despite the widespread usage of BPMN for process automation, EPC is accepted as the quasi-standard for process analysis in the business domain [92], [93]. When BPMod notations are evaluated in terms of workflow pattern analysis, BPMN has higher expressiveness compared to EPC, 11 patterns are not supported by BPMN and 31 by EPC [67]. However, it is found out that BPMN core and extended set covers similar constructs according to BWW model, while redundancy and complexity is increased in extended set [94]. In another study, the difference in understanding BPMN and EPC models are found to be insignificant [95]. Thus, studies support that increased number of constructs in a notation does not enhance the expressiveness, especially when we consider that our aim is to conduct BPMod to enable communication between users.

For making our choice as EPC, we consider that BPMN is evaluated to be the most complete notation, but it brings complexity and lack of clarity problems [34]. To overcome the problems modelers usually use a restricted set of constructs for business analysis [96]. BPMN with the restricted set and EPC indeed have similar expressive power for descriptive modeling. It is shown that EPC models can be converted to BPMN without a significant loss [97], [98], and transformation is supported with the tools [99], [100]. Thus we select EPC as it natively supports process analysis and modeling in the business domain. But we believe that BPMN is also suitable for our purpose, and plan to integrate BPMN as the alternative notation in the future.

2.4. Business Process Modeling Methodologies and Tools

The notations mentioned in section 2.2 are leading BPMod notations [34], [101]. These techniques and notations are commonly utilized in BPMod frameworks, methodologies, tools and approaches. Although there are various BPMod techniques, methodologies providing comprehensive procedures and guidelines are limited [34]; thus modelers have difficulty in choosing from various notations and applying them [102].

BPMod tools provide a software environment for modelers to develop model diagrams. They are usually more comprehensive than just a drawing tool. These are named as "suite", providing many other functionalities to the users including model object databases, model checkers, various modeling techniques, guidance for users, creation of reports, support for customization and process execution environments. Frameworks have also different properties, like describing the process of BPMod and defining a taxonomy for organizational artifacts.

One of the most common and well-known BPMod methodology is ARIS [50], [56], [103]. ARIS provides a modeling approach integrating different perspectives expressed as "ARIS House" as shown in Figure 11 and a modeling tool by "ARIS Collaborative Suit". ARIS supports a number of modeling perspectives and notations. This variety, rather than guiding the user for different modeling purposes, creates difficulty in selection and gives rise to inconsistent models with objects used in different senses by different roles. Additionally, as relations are not well-defined between perspectives and modelers are not guided with respect to their roles and goals; there are practical difficulties in modeling. There are many other BPMod tools available in the market, while ARIS is declared as a leading tool in terms of its comprehensiveness and usage [104], [105].

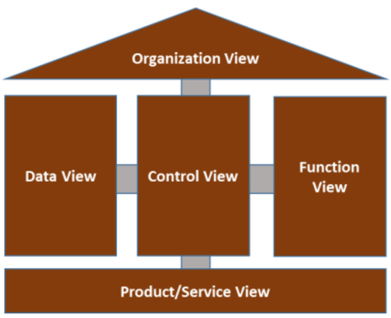


Figure 11 ARIS House

Other than ARIS, most commonly known BPMod frameworks focusing on enterprise modeling can be listed as Zachman Framework [106], CIMOSA Framework [107], GERAM Framework [108], DODAF Framework [109], TOGAF Framework [110]. Each of these frameworks has significantly contributed to process modeling activities and disseminated the usage of BPMod in different fields. However, they have deficiencies for practical usage in organizations and utilization of models for different goals. For example, Zachman Framework provides an ontology rather than methodology. It does not provide the necessary environment to develop the process models. CIMOSA and GERAM are specific to computer-aided manufacturing field. Although DODAF provides an extensive methodology, it does not focus on organizational level processes but focuses rather on complex operational level processes of military systems. Process oriented enterprise modeling (POEM) methodology mainly describes aspects like organizational structure, roles, business goals, relation between processes and business goals; rather than providing BPMod technique [111]. EPC modeling methodology, using EPC as the underlying notation, provides principles for modeling focusing on business process redesign [112]. PLURAL method, although providing detailed guidelines, is focused on decentralized process modeling in organizations [1]. There are other methodologies mostly based on BPM suites and focusing on process automation and execution, such as Bizagi [8]. BPMod techniques that include instructions on how to use the notations are limited in number and they have certain weaknesses [4].

In practice, the usage of “modeling tools” rather than modeling methodologies are encountered more frequently. BPMod is a field that is closely related to practical application, and very much supported by software tools. Resulting from the needs, there are many tool alternatives in the market, and many tools and plug-ins developed by academic initiatives as well. Usually, although not formally documented, a BPMod tool provides a framework for the user on how to conduct modeling.

The tools focusing on business process analysis (BPA) are related to the scope of this study. Gartner defines BPA as “*the business modeling space in which business professionals and IT analysts collaborate on business architecture, transformation and improvement, including process analysis and design to support BP improvement initiatives*” [113]. Furthermore, BPA can be a bridge to improve the alignment of IT efforts with business initiatives. As we didn’t encounter tools that directly supports diverse goals (like requirements engineering, software size measurement), our focus is on generic BPA tools; but we have given special focus on extended functionality provided by these tools (like documentation) so that we can gain insight on extended usages of BPMod.

We observe EPC and BPMN as the most frequently used notations in the tools. Although it is mentioned that EPC is the most popular and common process modeling technique [74], [75], [92], the number and availability of tools that support EPC are low. This is an opportunity to develop a state-of-the-art tool that supports EPC modeling to fill the gap.

Signavio is an online modeling tool that supports both BPMN and EPC notations [29], [114]. It supports process hierarchy and semantic checks. It also enables the definition of terminology within a dictionary, adding explanations and attaching documents to the objects. As it is an online tool, it has strong team collaboration capabilities and can publish the processes in the organizational portal. It can also be used to generate process documentation.

Other tools supporting EPC notation are Visual Paradigm Process Modeling Tools [115] and bflow* Toolbox [116]. Visual Paradigm tool enables modeling of separate but associated as-is and to-be models, then comparing them. It also involves generation of RACI chart, central repository and glossary management. bflow* Toolbox is an open source academic tool with significant features like live validation, batch modeling with the keyboard and the addition of new features easily by means of Eclipse environment.

EPF Composer is focused on not only graphical representation of processes, but also natural language statements [117]. It stores the diagrams in a process repository. But EPF Composer supports a restricted set of diagrams and notations and does not provide a methodology for modeling. The other example, MS Visio is essentially a visual drawing environment. It has gained popularity by means of its ease of use, integration with other office tools and low price. But in essence, it has nothing specific to a modeling methodology; and visual models are just graphics rather than semantically analyzable objects.

Among the tools supporting BPMN, Bizagi Process Modeler supports validation and extended object attributes [118]. It can be used for process document generation. iGrafx Process tool supports many other diagram types together with BPMN. It enables resource modeling, what-if analysis, simulation and reporting [119]. Sparx Systems Enterprise Architect is originally a software design tool, but supports BPMN modeling [120]. Cheetah Experimental Platform is also an academic BPMod tool which is used to conduct experiments on the process of BPMod [121].

2.5. Usage of Business Process Modeling for Different Goals

Business process analysis is conducted for various goals, and business process models are utilized for various purposes both in enterprise BPM and in software development. However, the studies for different goals are handled separately and different methods and notations are developed for them in the literature. For example, for process improvement notations like EPC that provide better understanding on processes are utilized; while formal notations like BPMN, BPEL, Petri-Net (Peterson, 1981) are utilized for process execution and automation. There are a few studies that directly relate business process models to software [18], but widely accepted approaches have not been come into usage yet. In this section, we analyze the studies in the literature in the diverse areas for which we specified for UPROM.

2.5.1. Business Process Modeling for Requirements Engineering

Requirements engineering deals with identifying the requirements that clarify the desired outcomes as the client wants, and defining specifications to describe how the proposed system will work [42]. Requirements are the hardest artifacts to materialize, while they are also the ones most effective on the success of the project [42], [122], [123]. Unstable requirements are found to be one of the main reasons of project failures [124]. Requirements engineering activities in SDLC covers analysis of requirements in different levels and perspectives, and also of different types. A requirement is [27]:

- A capability that a user needs to achieve an objective.
- A capability that a solution owns to satisfy a specification.

Business requirements describe the high level objectives of a project and the rationale for initiating the project [27]. Stakeholder requirements, or user requirements explain needs of different types of stakeholders. They identify the interaction of those stakeholders with the solution. User requirements establish a bridge between business requirements and system and software requirements [27]. Software requirements describe the functionality expected from the software in detail and in software terms.

The requirements are divided as functional and non-functional. Functional requirements explain the functionality and capability expected from the solution in terms of behaviors and operations [27]. They include functionalities for transfer, transformation, storage and retrieval of data, especially for business application system type software [125]. Nonfunctional requirements are not directly related to the behavior of the solution. They can be quality constraints, organizational, environmental and implementation requirements [125].

The difference between business and system requirements is that, business requirements describe the value to be delivered to the business in business terms, while system requirements describe a high level design of the product [126]. Business requirements are often referred to as problem of the real world [127] or high level system requirements [128]. Goldsmith also emphasizes that defining business requirements and identifying system requirements based on business requirements is critical for success of the project. Otherwise, this ends up with requirements creep problem. The well-known software process maturity standard, CMMI, also defines separate practices for eliciting business requirements and then developing system requirements based on those [25]. Business process models are already a way of describing what is expected from the system. Our approach moves forward from business requirements and provides a bridge to system requirements by detailing the system still in business terms but focusing on data perspective and functionality in more detail. In this study, we aim at identifying user requirements from the business perspective by using business process models. We use the terms user requirements and system requirements interchangeably to specify the generated requirements by UPROM.

The amount of effort allocated to identifying requirements is varying too much among projects, thus hard to generalize [42]. A study suggests that 5% is spent on requirements not including specification [129], and another study measured 16% on all requirements engineering activities for 15 projects investigated [130]. These show that requirements engineering activities cover a high percentage of effort in total SDLC and affects the total cost of the project, as requirements defects are the defects that require the most effort to fix [41].

BP models are used as a method for requirements elicitation, and "documenting and codifying" business process requirements [15], [123]. BPMod is used as a common tool to collect requirements during software development [5], [12], [14] and seen as an essential part of those activities [92], [131]. There are even studies to assess the expressiveness capability of business process models for software requirements [123].

Models to identify the requirements are used for quite a long time [132]. But current BPMod languages and tools do not directly support an integrated approach for requirements analysis activities [17], [133]. For the existing ones, they are not sufficient to define user requirements in a structured and traceable way to explain how the automated system is supposed to behave to conduct each activity in business processes [16]. Due to the importance and intangibility of the requirements, it is critical to utilize information captured on other activities effectively to identify the requirements. Furthermore, organizations require numbered and traceable requirements statements for scope management during project management, reviewing by different stakeholders and subcontracting [134]. Literature study by Nicolas and Toval concludes that although there is value in generating textual requirements from models, approaches do not exist to form requirements sentences from business process models [16].

Nicolas and Toval presents the results of their literature review study on the generation of requirements specifications from software engineering models [16]. This study aims at identifying any study that proposed a method to generate requirements specifications in any format. An important outcome of this paper is that, it states "*no approach exists to address the issue of maintaining synchronization between the documents or requirements generated and the initial models*" [16]. This is also a problem identified and aimed to be solved within this study. Within this extensive review, the study of Türetken, Su and Demirörs is presented as an important study to form requirements from business process models [18]. This is a previous study of our research group in military domain that showed positive results on the topic. We utilize the experiences obtained in this study. A recent study of our research group also produced results in government domain and utilized as input for this study [19].

The study of Berenbach et al. is also important to associate UML models of processes and requirements [135]. Daniels and Bahill also suggest that use cases are important for requirements [136]. Mayr et.al. developed a specific simple language to conduct requirements modeling while developing business process models at the same time.

These studies are valuable to provide a basis for using models for requirements, but are not related to our study as they are not based on BPMod notations [14].

Cox et.al. proposed an approach for deriving and contextualizing software requirements from business process models through the use of the problem frames approach [17]. The triggering point is similar to our study. They mention that while it is agreed that process models are valuable in requirements engineering, how process model maps to related requirements activities is not clear. Cox et al. claim that the problem 'framing' should suggest appropriate notations for requirements analysis and specification. If problem frames for complex situations handled by process models can be formed, then these problem frames would be used within requirements phase. The study includes an application in the industry. Although it is an extensive study, results are appropriate only for specific cases. In their book, Berenbach et al. emphasizes the importance of associating requirements engineering to business processes [135]. Business processes are implied to be identified by use cases. This is an important study as it provides insight on how to handle business processes in a holistic way to reach at good requirements.

Staccini et.al. developed business process models using SADT and IDEF0 languages and elicited user requirements by using these models for health care processes [137]. After identifying the steps of a process, they analyzed each step to specify the required data entities that are added, retrieved or modified during that step. They also bound the data necessary to calculate process indicators. Then, focusing on the entities, they deduced data flows for the sub-activities and the related actor together with object oriented diagram of a process data model. This study reveals the need to identify requirements based on the data entities utilized in the processes, modeling the data entities together with business process modeling. However, it is developed on unstructured models and manual operations.

The study of Specht et.al. is an important example of how business process models can be customized and the objects of a business process model can be used directly to form some other artifact, BPEL code for service oriented architecture [133]. Although the aim of using process models is different, closing semantic gap between business process layer and IT layer, the idea of using models is similar to this study. They aim at reaching an output of distributed service oriented IT architecture and workflow orchestration by using process models. EPC and FAD notations are used to model the processes. They are extended to embed the related information.

We observe that the viewpoint shifted through "goal-oriented" approaches [122], [138]. Goal oriented analysis is especially beneficial in early phases of requirements analysis and to reveal non-functional (or quality) requirements. GRL is among the most popular goal oriented language among other languages like KAOS, i*, NFR and TROPOS [139]. The idea of goal oriented analysis is, at the initial stages of requirements analysis, the customer can identify non-functional requirements expressed in terms of "soft goals" [138]. During this analysis, particular functions of the system will also be identified to determine how those nonfunctional requirements will be achieved. In this way, nonfunctional requirements that are expressed as soft goals and high level functions of the system are identified during goal oriented analysis. Naturally, soft goals and functional goals will be conflicting with each other, as for example flexibility goal will clearly conflict with cost-effectiveness goal. The models drawn as a result of this analysis serve the purpose of revealing and analyzing those kinds of conflicts and alternatives to solve them [139]. Although goal oriented approaches are good examples of integrating BPMod and requirements analysis, they are mapped to very early stages of development. Thus the phase they are used are far before descriptive analysis and modeling of BP models in SDLC.

URN is a language developed for requirements engineering activities, used to discover and specify requirements while analyzing correctness and completeness of them [139], [140]. One of the aims of URN models is specified as analyzing business processes while discovering and specifying requirements. The perspective of URN is not the detailed specification of functionalities to be supported by the system, but rather the reason of choosing related behaviors together with a high level identification of the capabilities and architecture. With this approach, URN has a similar point of view for UPROM studies, focusing on user level functionalities to identify high level capabilities. URN utilizes two different languages: Goal-oriented requirement language (GRL) that is used to model actors and their intentions; Use Case Maps notation to describe scenarios and related architectures. i* notation, which is one of the first examples of goal-oriented requirements engineering approaches, dates back to the nineties [132]. i*framework and notation propose analysis and modeling solution for early phase requirements engineering activities that come before the initial requirements formulation phase.

The notation of URN combines GRL with UCM notation for modeling scenario concepts [139]. The scope of URN covers the specification of behavior, structure, goals and nonfunctional requirements of a system [141]. Weiss and Amyot emphasize that all of these aspects are claimed to be relevant for BPMod and URN can meet the objectives of BPMod. In their study, they developed business processes and identified web service definitions automatically using these processes. Then, they utilized this information to depict service provisioning relationships between components as a UML deployment diagram. They moved on to generate message sequence charts automatically from the scenarios. They also suggest that this study can be complemented with validation approaches like reviews by stakeholders, generation of test goals and additions of performance annotations. We observe this study as a good example of business process modeling and usage of process models in an integrated way for multiple goals, although the goals and the analysis phases are different.

REDEPEND generates excel sheets from i* goal models [142]. The goal models rather focus on non-functional requirements and high level functions of the system in early stages, as described in URN section. Thus, this tool provides a description of early and high level requirements of the system. The other tool, Objectiver, also identifies early and high level requirements based on goal models and generates documentation based on descriptions of goals and high level functions [143].

PHILharmonicFlows framework focuses on the need for object-awareness of processes with process management systems [144]. Micro processes include object types, attributes and state transitions of single objects, while macro processes depict state transitions of different objects and relations between them. Other than processes, a relational data model of the object types together with attributes and relation types is developed. PHILharmonic modeling methodology follows three steps: "*stakeholders elicitation and domain data modeling, behavior and functional requirements modeling, rapid prototyping*" [145]. At first step, object types and relations are modeled on a data model and user types are defined. Data centric modeling of processes in two levels of granularity aims to provide processes in tight integration with data. The researchers name these process models as "functional requirements modeling", as it describes the behavioral perspective of the software system [145]. This approach proved itself to automate modeled processes as efficient workflow systems [144]. However, considering the concepts used in process modeling like process and object instance constraints, object attributes and their changes, user authorizations, we believe that these requirements models are closer to the technological side of business process modeling. Thus, earlier analysis of business processes and user requirements are necessary in previous stages. Such models shall be developed based on the knowledge captured and come to an agreement with domain experts.

Vara et.al. utilizes a requirements engineering approach to elicit and specify requirements from business process diagrams [131]. The first step of their approach includes analyzing business process diagrams to elicit functional requirements of the PAIS. The first stage models to be developed are glossary, business events, domain data model (entities and their relations), business rules, a role model and a process map [146]. Business process diagrams are developed based on these models. Then, each element of diagrams are labeled to be automated or not. For the tasks labeled as to be automated, textual descriptions are entered based on a use case template; and some of the fields of this template are filled from the models. We observe that most of the information provided in this template exists in the process definition document we generate, like input, output, role, pre and post condition and business rules. This study is close to the technological side, analyzing data in detail for process automation. However, it is a good example and provides us insights to integrate BPMod and requirements analysis.

It is important to identify the distinction between requirements and specifications. While requirement expresses the desired property within the environment, a specification identifies the behavior of the system on the interaction with the environment [147]. Initially, a requirement is identified, and a specification is derived using it. Specifications can be directly utilized to develop the software [147]. In this study, our focus is on establishing requirements on a user's point of view, by utilizing business process model information. The requirements focus on identifying the "goals" of the system to be developed from a user perspective [122]. Though models of UPROM capture some information on how the system will work, more analysis is required to turn them into structured software requirements specifications.

Considering the number of studies in the field, we can conclude that BPMod and requirements engineering fields are in tight relation. However, the studies focusing on utilizing BP model knowledge systematically for requirements engineering are restricted in number.

2.5.2. Business Process Modeling for Software Size Estimation

Software functional size estimation (FSE) is essential for time and cost estimation in project planning [23]. For functional size measurement of a software system, a complete set of functional requirements is necessary which specifies data movements for each functional process [125]. This is characteristically achieved when software requirements specifications are developed, which means reliable time and cost estimation can only be conducted after this phase.

There exist a number of methods for functional size measurement (FSM) of the software. COSMIC FSM method is amongst the most common and accepted ones [125]. COSMIC is applicable to the software domains of business application software, real-time software and their combination. It was accepted as a standard by ISO in 2003 [148]. The unit of measurement for COSMIC is COSMIC function point (CFP), which is one unit of data movement (DM). The total functional size is calculated adding up all DMs identified by using software requirements.

Marin, Giachetti & Pastor (2008) conducted a comprehensive survey on the measurement process approaches that use conceptual models [149]. This survey indicated eleven proposals of functional size measurement procedures based on COSMIC FSM. This survey has been one of the most important and detailed surveys on the measurement process proposals so far. It reveals that UML models are the ones most commonly used for size measurement. Other studies we examined also utilized UML models [150], [151]. Some of the methods provide a descriptive procedure and do not have a formal foundation [74], [152].

There are studies in the literature utilizing other artifacts to determine the functional size of the software in an automated way. Lamma et.al. utilized ER and DFD diagrams to conduct IFPUG v4.0 size estimation [152]. Use cases and sequence diagrams are most frequently used artifacts. Jenner (2001) utilizes use case and sequence diagrams [153], Levesque and Bevo take sequence diagrams with use cases to generate COSMIC FFP v2.0 estimation [154]. There are other approaches that utilize design artifacts like object oriented method artifacts [155] and RUP artifacts [156]. None of these methods provide early size estimation for the system.

In their study Condori-Fernandez et.al. utilized use case and sequence diagram artifacts of requirements analysis to estimate the COSMIC functional size [155]. Connections between use cases and messages on sequence diagrams are used to identify data movements. Their approach is similar, as the identification of data movements is based on input, output, creation, update and destroy relations on sequence diagrams and refined with further rules. However, the approach does not cover BPMod and sequence diagrams require a more detailed level of requirements analysis.

In summary, we observe that there are studies that utilize models to identify the functional size of the software, but they are mostly based on UML and other models. There are a few studies associating BPMod and FSE. Monsalve et.al. proposed a method to measure the software functional size from process models [23]. However it identifies data movements directly on process models, which requires a size estimation expert to conduct the analysis. Our previous studies, used as input for this study, also provided fruitful results on the size estimation using process models [22].Our group also extends the size estimation studies to be applied in S-BPM domain.

Early function point analysis is dependent on the estimator's ability to apply the related method [20]. Thus, user requirements shall be expressed as formalized as possible to prevent subjectivity [157]. COSMIC Method Manual is specially extended for business application software, which is defined to "*have the purpose of capturing, storing and making available data about assets and transactions in the business world*" [158]. This purpose matches perfectly with the application software to be developed to automate business processes. Existing approaches utilize COSMIC Function Point Size Measurement Method to estimate the size of the application software to be automated. In order to decrease the subjectivity of early size estimation methods, definition of requirements has to be formalized. Since business process models enable the identification of software objects at higher abstraction levels, they are very useful to create a structured estimation process and a standard guideline.

2.5.3. Business Process Modeling for Process Documentation

Business process documentation and business process improvement are found as some of the most important purposes for which conceptual modeling is conducted in organizations. Many international standards and reference models require process documentation [24]–[26], [159]. Some of them even defines what is expected from the documents and their typical content [160].

ISO 9001:2008 maps an output of "quality user manual" to be essential as the output of defining the processes. It also provides the contents of the manual; the scope of the quality system, the procedures utilized in the quality system, detailed business process descriptions and relations between them. A capability maturity reference model, CMMI, identifies what information a defined business process includes; purpose, inputs, outputs, initial conditions, activities, roles, rules, verification, outputs, exit conditions.

ITIL and COBIT indicate that process definitions shall be documented as process books and reviewed periodically. ITIL provides a detailed process documentation template including elements like process name, description, purpose, inputs, outputs, products to be delivered, roles, start and end conditions. Process definition document is an important deliverable also for software development process. The process definition document is seen as essential by many stakeholders either together with models or alone.

The studies indicate that it is necessary to use natural language texts to validate the models and obtain feedback from different users [161], [162]. Studies in the literature for generating process documents from models are limited and mostly focused on conceptual models [161]–[163]. Another study focused on generating natural language phrases to textualize control flow properties of the activities [164]. Some BPM tools also have capabilities to generate documentation like in where process elements are listed without any grouping and cross process relations are not considered [29], [118].

Defining a business glossary or "business term catalogue" is recommended in a business process definition study, especially if the processes are utilized by a wide variety of users [2], [27]. Business glossary containing the domain terminology is required to provide a common understanding among stakeholders. There are tools that can be used to define terminology and obtain a list [114], [115], but obtaining a business glossary is not stated as part of a methodology. Business processes include many terminology, but it is especially critical to achieve unique and agreed definitions of the terminology.

Business process improvement is inherent to BPMod activities. A detailed method of BPI is to define as-is processes, then identify improvements on it and model to-be processes. An example tool allowing such a method is found at [115]. In the cases where organizations model as-is and to-be processes separately, or they conduct all activities on the same set of models, they require a mechanism to record the improvements they identified. This is especially required to present to the high level management to make the benefits of BPMod studies visible. Existing BPMod tools are specified to be unable to support recording of improvement suggestions [165]. Those improvement aspects are hard to identify after BPMod is completed, as the analysts are not submerged into the models. We didn't encounter any specific way of recording and reporting improvements in existing methodologies, other than placing notes on the diagrams.

2.5.4. Business Process Modeling for Process Metrics Definition

With the rising popularity for automated process execution, the terms like process performance management, performance monitoring, business activity monitoring have come into frequent usage [32]. Process performance measurement is referred to as determination of business process performance by means of some performance indicators. Process monitoring or process performance monitoring is the continuous follow-up for the process executions in an organization. Business Activity Monitoring (BAM) is defined by Gartner Group as "*processes and technologies that enhance situation* *awareness and enable analysis of critical business performance indicators based on real-time data*" [51]. It is described to be implemented by many kinds of software tools. Collecting key performance indicators is seen as a prerequisite for a complete achievement of process management [166].

The aim of performance management is to provide the organizations to run in alignment with organizational goals. Business performance management is conceived in conjunction with BPM, which can be defined as "*company performance management through processes*" [167]. However, there are not extensive studies to build up a performance management infrastructure of an organizational goals and business process models. One of the approaches for this is Scheer's Advanced BPM Assessment, in which Sheer indicates that business process management and business performance management are two topics going hand in hand. The second approach is of Thiault, which advises business performance management a method applied through processes that helps to improve organization's performances in a changing and complex environment [168]. Another collection of information in the field is from Taticchi, which is focused on business performance measurement and management [169]. The collection of experiences he provided focuses on sectorial applications, providing some insight on using processes from time to time.

From our literature studies, we observe that performance management is a popular concept that is mostly seen to be related to business process management and utilized in automated workflow systems, which aims to provide the users with metrics collected from run-time data. KPIs defined during descriptive BPMod activities are usually abstract and do not specify how the metric can be defined and collected in the automation software. We have not encountered an infrastructure to focus on guiding organizations to define process metrics which are then can be used as input to process automation.

Key performance indicators (KPI) (or metrics, which are used interchangeably with KPI) are found to be a key instrument to detect the state of the processes and to identify undesired behavior [170]. There are many tools that support the definition of KPIs based on the business processes and collecting data based on these KPIs. Process Performance Manager (PPM) by IDS Scheer is one of them. It enables the definition of attributes related to EPC process models. It is critical to define the connection between the process and its key performance indicators, otherwise it is hard to end up with meaningful results for process monitoring [166].

The usage of KPIs for performance management purposes involves four steps: definition, measuring, analysis and report [170]. As explained above, the definition of KPIs related to process models and the mechanisms to collect related data during the execution of those processes are much studied on the academy and industry [32]. The calculation and evaluation methods of the indicators and the connection of KPIs to processes must be defined clearly [166]. However, there are not many studies on defining KPIs in a structured way during descriptive BPMod to be used as input to process automation. Thus, we cannot find much guideline on how to determine KPIs to evaluate the performance of business processes. Kronz in his study mentions that definition of KPIs must be integrated with business process models [166]. Kronz defines “measurement point” as the point in each process execution to collect data related to a KPI. This point must be applied by the application system to be developed. He explains that, within ARIS methodology, the data points for a KPI is defined by means of a KPI allocation diagram. At their enhanced notation, the measurement point, an event, is marked on an EPC diagram. The data to be returned is stored in ERM attributes.

As we claim, Kronz also emphasizes that the definition of measurement points, related data to measure the KPI and calculation rule for a KPI, even if not mathematically expressed at this stage, constitute the requirements of the information system to be developed for the processes [166]. We agree with Kronz to integrate business process models and KPI definition, but believe that a more integrated definition of them is possible. KPIs can be grouped into two types: [32]

- Metrics related to process execution and do not require business-specific data (like system load, number of concurrently running processes)
- Metrics that require business data (like number of project applications collected in one term)

In this study, we refer to group one KPIs as metrics related to system performance and exclude them from KPIs determined in process metrics list of the organization.

An extension is proposed to URN notation [140] to define indicators and monitor performance of business processes [171]. In their methodology, KPI is defined as the measure of the satisfaction of business goals and performance requirements of a system. The required performance indicators and metrics are defined on modeled business processes. Each of the KPIs is attached to the data warehouse to identify the storage from which information will be gathered for KPI. Pourshahid et.al. integrated the business process models with business intelligence and business monitoring tools to monitor KPIs and find improvement opportunities [171]. KPI symbol within GRL metamodel is defined to include the following attributes:

- Boolean attributes to indicate if a KPI is a time, cost, quality or flexibility measure,
- Target Value, threshold value and worst value,
- Data source for value and report.

KPI symbols are attached to the goals which are also attached to tasks on a GRL diagram. In this study which implemented related KPIs with a business intelligence tool and monitored them, we observe the necessity of defining projected values and data repositories to collect and report the values.

In summary, although business performance management is a popular concept in BPM field, similar to the requirements engineering field, we observe a need to integrate business and technological aspects of an organization by defining KPIs in descriptive models that can be used as direct inputs to define metrics in process automation. In BPM notations, metrics are defined in abstract level, without details for measuring [56].

CHAPTER 3

THE UPROM METHODOLOGY

The development of business process models that can be used to generate artifacts for different goals requires a comprehensive solution to guide users in achieving the related results. In addition to the notation, the set of activities to analyze the business processes and functions for process automation, add the required information to the models and generate the artifacts must be described.

This chapter presents the proposed unified BPMod methodology, UPROM. The components of the methodology were depicted in Figure 2 in Chapter 1. In section 3.1, the notation of UPROM is described in detail. This section also provides guidelines on the usage of notation elements, including diagram types and constructs. Section 3.2 covers the unified BPMod process and guidelines applicable for the whole modeling project. Section 3.3 explains the generation procedures for the artifacts. In section 3.4, UPROM tool developed as a prototype to apply UPROM is presented.

3.1. The Notation and Guidelines on Using the Notation

The notation of UPROM comprises six diagram types, all of which are interrelated with a common meta-model. EPC is the core diagram of the notation to model the control flow and behavioral perspective [56]. The diagrams as part of the notation are as follows:

- Extended Event Driven Process Chains (EPC)
- Value Chain (VC) Diagram (VCD)
- Function tree (FT) diagram (FTD)
- Organization chart (OC) diagram (OCD)
- Entity relation (ER) diagram (ERD)
- Function allocation (FA) diagram (FAD)

Each of these diagrams has distinct focuses to analyze business processes and the functions to be automated. The usage of these diagrams in UPROM are summarized below. Each of the diagrams serve the purpose of representing a different perspective of business processes, as shown in Figure 12.

- **EPC:** It is the core diagram to conduct business process analysis.
- **VCD:** It is utilized to grasp an overview of the whole system together with the risks and objectives. High level processes are organized and the relations between them are depicted by this diagram. Value chain symbols in VCD are detailed by other VCD, EPC or FTD in a lower hierarchy.
- **FTD:** It is used to organize processes which are not in a flow relation but interrelated. Usually, the functions in FTD are detailed by EPC.
- **OC:** It is the diagram that covers all organizational elements in EPC, FTD and VC diagrams. It aims to identify the relations between organizational elements.
- **FAD:** It is the main tool to conduct function allocation analysis for the functions to be automated. Requirements and functional size estimation are basically generated using the information in FADS. FADs are assigned as sub-diagrams for the leaf functions in EPCs.
- **ERD:** It is used to depict entities placed on FADs and the relations between them.

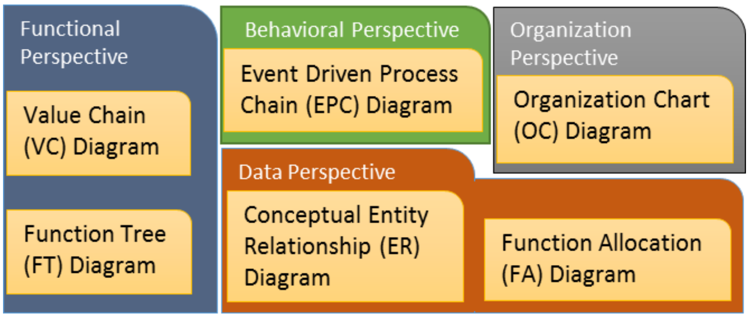


Figure 12 UPROM diagram types and related modeling perspective

The Design Rationale for the Notation:

The business process models developed in UPROM should serve the need for detailed descriptive business process analysis and should enable the analysis activities for other purposes of UPROM including requirements analysis, software functional size estimation, process documentation and process metrics identification. In order to comply with this need, the modeling notation should be usable and understandable by the business experts and enable further analysis options.

We analyzed the literature for available notations and provided the results under Chapter 2. Each of the notations has different advantages and disadvantages in terms of usability, extendibility and flexibility to be used for other purposes. Utilizing the existing domain knowledge, we developed a notation that can be used to model different process perspectives. With this notation, we aimed to grasp information on business processes and functions to be automated in a structural manner and from the business perspective.

Curtis et.al. defined the need for representing different perspectives for process models including functional, behavioral, organizational and informational perspectives [37]. As one of the guidelines for BPMod, Becker et.al. also suggests the usage of guideline of systematic design for the integration of business process and information models [2]. In UPROM, we integrate different perspectives that are necessary to build the bridge from BPMod to process automation with the six diagram types. In the modeling environment, all diagrams are developed in an interrelated structure within a modeling project.

There are various factors that affect the understandability of business process models [172]. Mendling and Strembeck revealed that "personal, model and content related factors" has the most effect. We consider such factors by providing a modular methodology for BPMod goals. By means of modularity of the notation and process, the user can choose to apply some of the UPROM aspects depending on her goals.

Assuring the syntactic and semantic correctness of models is also one of the guidelines of modeling. However, treatment of correctness can change with respect to the aim of BPMod. The potential perspectives focusing on different aims include workflow specification, simulation, human resource planning, project management, knowledge management, benchmarking, certification, software development [2]. Process models that provide communication with end users may have less strict conformance to syntactical rules, if this provides compact and clear methods to enhance understandability [2]. UPROM's aim is to analyze processes from a user perspective and to come to an agreement with the user on both business processes and requirements for process automation. Thus, we keep this fact in mind as we define the modeling methodology, and give special attention to provide a notation understandable from user and business perspective rather than a notation that aims to strictly conform to formal semantic and syntactic rules. Another aspect we focus on the notation design is that, although the notation supports analysis in the business domain, we capture the process information in a structured way so that it can be transferred to other artifacts.

We described our rationale for selecting EPC in section 2.3. We especially focused on a comparison between EPC and BPMN, and concluded that EPC is more suitable for our purposes, as we especially focus on business domain. However, with its restricted set used for analysis, BPMN can also be utilized in a similar manner within our approach.

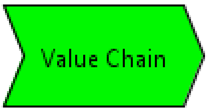
Placing EPC in the center and representing different perspectives, the notation of UPROM has commonalities with ARIS framework [173]. In contrast to the abundance of diagrams and attempt to fulfill various purposes in ARIS, UPROM has a limited number of diagram types and guides the user specifically for structured descriptive modeling and transfer of knowledge to automation. FAD, different than ARIS, is used for allocating and analyzing the functions for automation.

In the following sections, we describe diagram types and related constructs. For each diagram and construct, detailed descriptions and guidelines on the usage are provided. The "note" construct is available for all diagrams and can be connected to all constructs. All constructs have the following attributes assigned: Name, ID, description. Some of the constructs have additional attributes as will be mentioned in the following sections.

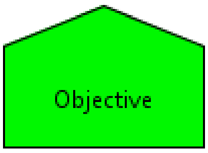
3.1.1. Diagram Type 1 – Value Chain Diagram (VCD)

Value chain is a concept encompassing all processes in an organization to deliver value as products and services to the customer, providing a business overview [15]. VCD is also a part of the ARIS methodology to model process view [173]. VCD is an optional diagram in UPROM to model high level processes in an organization within the scope of the modeling project. It models the processes from the functional perspective. It is used to depict high level processes and the relations between them. Thus, it is usually utilized as the "process map" of the modeling project, which is the highest level functional and behavioral diagram file in the modeling project. The construct that can be placed as objects on VCD are listed below. In UPROM, we use VCD notation the same as the existing definitions in the literature [173]. As the detailed usage of these constructs and possible relations are not described in detail in literature, it is important as part of our methodology to define the semantics of those constructs, explaining how those constructs represent real life objects.

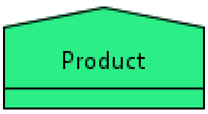
Value Chain: High level process modules that add value to the creation of products/services. ***Additional attributes:*** sub-diagram.



Objective: The high level objective that is aimed to be achieved by the connected value chain. ***Additional attributes:*** link.



Product: The output (as product or service) that is to be achieved by the related value chain. As value chain objects represent high level processes, the output of them are also shown in high abstraction level representing the product or service. ***Additional attributes:*** technical term, link.



Risk: The high level risk associated with the connected value chain. ***Additional attributes:*** technical term, link.

Possible connections:

- **Value Chain – Value Chain:**
  - **is process-oriented superior**:
    Directed connection.
    The object on the line side is the process oriented superior of the one on the arrow side; encompassing the one on the arrow side.
  - **is predecessor of:**
    Directed connection.
    The object on the line side is the predecessor of the one on the arrow side; thus completed before the one on the arrow side.
- **Value Chain – Objective:**
  Undirected connection with no arrow, both directions having the same meaning.
  "1 to n" connections are allowed in both directions.
  The value chain achieves the connected objective.
- **Value Chain – Product:**
  Undirected connection with no arrow, both directions having the same meaning.
  "1 to n" connections are allowed in both directions.
  The value chain produces the connected product or service.
- **Value Chain – Risk:**
  Undirected connection with no arrow, both directions having the same meaning.
  "1 to n" connections are allowed in both directions.
  The connected risk can occur during the execution of the value chain.

Possible sub-diagram assignments:

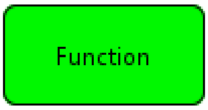
- **Value Chain:** can have another FTD or EPC assigned as sub-diagram.

3.1.2. Diagram Type 2 – Function Tree (FT) Diagram (FTD)

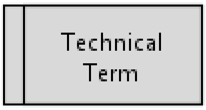
FTD is used to depict the decomposition of a process into lower level processes which do not have control flow relation between each other, thus typically the processes not following each other in a control flow. Still, there can be implicit predecessor – successor relations between sub-processes due to event and input/output relations in EPCs. Due to its nature, FTDs are usually placed in higher levels of the modeling project hierarchy, but can also be used in lower levels. It is used in a similar way or as an alternative of VCD.

FTD exists also in ARIS framework, however it is used as an alternative static view of an EPC, placing all functions represented in an EPC and showing relations between them [173]. In this style, FTD is only a derived diagram from existing EPCs. In UPROM, we utilize FTDs to model functional decomposition of a high level process that are not in a control flow but compose parts of a higher level process module. We also add a "technical term” construct to FTD so that the terminology related to the functions in FTD can be defined explicitly.

Function: A task or an activity performed as part of a business process. All functions in FTD must be detailed in another FTD or EPC, thus they cannot be leaf functions. ***Additional attributes:*** sub-diagram.



Technical term: A term in the related domain. The definition of the term is provided in “technical term” attribute. An abbreviation can be defined for the term in “description” attribute. ***Additional attributes:*** technical term.



Possible connections:

- **Function – Function:**
  Directed connection.
  The object on the line side is the process oriented superior of the one on the arrow side; encompassing the one on the arrow side.
- **Function – Technical Term:**
  Undirected connection with no arrow, both directions having the same meaning.
  “1 to n“ connections are allowed in both directions.
  The function description involves the connected technical term.

Possible sub-diagram assignments:

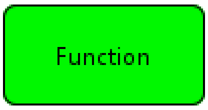
- **Function:** can have another FTD or EPC assigned as sub-diagram.

3.1.3. Diagram Type 3 – Event Driven Process Chain Diagram (EPC)

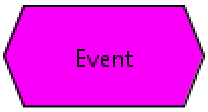
EPC is in the center of the notation of UPROM. EPC is used to analyze and model control flow, thus basically represents the behavioral and functional perspectives of the business processes. Additionally, with the extended notation, EPC is also strong for organizational and informational perspectives. Standard extended EPC notation as defined by the ARIS framework is described in section 2.2.1. EPCs can be assigned as sub-diagrams, and each function in EPCs can be detailed as a process in another EPC. If there is no further EPC or FTD assigned as a sub-diagram for a function, that function is called a “leaf function”.

The list of constructs for EPC used in UPROM are listed below. The constructs that do not typically exist in EPC notation and added for UPROM, or those utilized with a different semantics in UPROM are mentioned below.

Function: A task or an activity performed as part of a business process. Functions transform the input resources to outputs. Functions are named imperatively, directly stating the activity to be conducted [9]. ***Additional attributes:*** sub-diagram.



Event: Event represents the state of the system in the business context. Events trigger functions and functions result in events, thus forming a chain. Events can be external changes, internal changes of state, or the final outcome of a process [173]. Start events, events without an incoming connection, are the "entrance criteria" for the process, and identify the conditions for a process to be initiated. End events, events without an outgoing connection, are the "exit criteria" for the process, and identify the conditions for a process to be completed. Events are named in "noun-verb" format, indicating the completed action or state of the system.



Logical Operators: They are used to model alternations and loops in the control flow. Together with functions and events, they can be utilized in many ways to organize the control flow as shown in Figure 4. If used in the middle of the control flow, they are placed as pairs forming a block structure. The first one of the pair is called split and used to initiate parallel or alternative flows. The second is called join and is used to gather and join alternations and parallel flows. They are used as single objects at the beginning and the end of the processes.

- ***AND operator:*** It is used to model two or more parallel flows.
- ***OR operator:*** It is used to depict decision points where one or more possible paths are to be followed.
- ***XOR operator:*** It is used to depict decision points where only one of the two or more possible paths are to be followed.

Process Interface: In ARIS framework, this construct is used as an "off-page connector", linking the start and end of continuing processes [173], [174]. In UPROM, we utilize process interface construct to enhance modularity by placing a reference to another process module. It is similar to a method call in an object oriented software code. The detailed semantics of using process interface construct is provided in the guidelines in section 3.2. ***Additional attributes:*** sub-diagram.

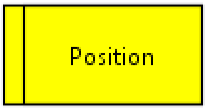
Business Rule: The object is used to state the rules that specify or constrain how the connected function is executed. They "*define, constrain, or enable organizational operations*" [27]. ***Additional attributes:*** link.

Application: It represents software tools and systems that support the processes to perform the connected function. In ARIS framework, it is called as "Application System". ***Additional attributes:*** link, technical term.

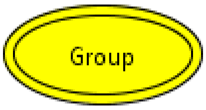
Organizational Elements: This is a collection of constructs representing organizational aspects of the processes. Organizational element constructs are listed below. ***Additional attributes:*** technical term.

- ***Organizational Unit:*** Department or permanent grouping performing certain functions in business processes.

- ***Position:*** Roles assigned to an individual person.



- ***Group:*** Temporary groupings of persons working together for a specific set of tasks.



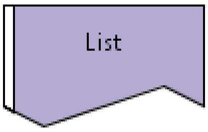
- **Location:** The physical location the activity is conducted.

- **External Person:** An external person entity for the organization.

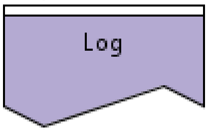
Information Carriers: This is a collection of constructs representing different types of medium to store data and information in the processes. They can be in hard copy or digital format. Information carrier constructs are listed below. There are many other information carrier constructs in ARIS tool, however we utilize a limited number of constructs to prevent complexity. ***Additional attributes:*** technical term, link.

- ***Document:*** It is the generic information holder used as input and output for the functions. It can be used to represent digital or hard copy objects depending on the modeling style.

- List: It represents a formal or informal list of a data.



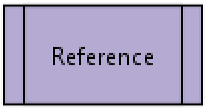
- **Log:** It is the data that is to be captured informally. When data on notes, descriptions or decisions need to be stored as an artifact of the process in an informal way, this construct is used.



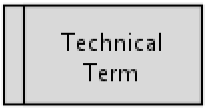
- **File:** It represents a digital medium that stores data. It can store batch data in a database or single group of data.

- **Product:** The output (as product or service) that is to be achieved by the related function. It represents the end product.

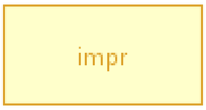
- **Reference:** This is an information carrier, but it is not transformed during the function execution, but it acts as a catalyzer. The construct is used for data or documents that are used as a reference for the function such as laws, legislations and process documents.



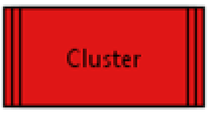
Technical term: A term in the related domain. The definition of the term is provided in "technical term" attribute. An abbreviation can be defined for the term in "description" attribute. ***Additional attributes:*** technical term.



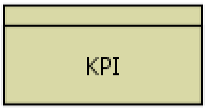
Improvement: It is a construct added specifically for UPROM. It is used to specify the improvements achieved (or to be potentially achieved) by business process analysis.



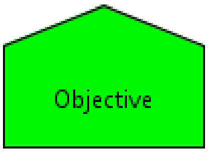
Cluster: It represents the collection of entities. ***Additional attributes:*** link, technical term.



Kpi (Key Performance Indicator): It is used to identify the key performance indicators for the business processes. ***Additional attributes:*** collection period and frequency, hypothesis.



Objective: The objective that is aimed to be achieved by the connected function. ***Additional attributes:*** link.



Risk: The high level risk associated with the connected function. ***Additional attributes:*** technical term, link.

Possible connections:

- For all connections between these constructs:
  - **Event – Function/Process Interface**
  - **Function/Process Interface – Event**
  - **Function/Process Interface – Function/Process Interface**
  - **Logical Operator – Function**
  - **Event – Logical Operator**
  - **Logical Operator – Event**
  - **Logical Operator – Logical Operator**
  - **Function/Process Interface – Logical Operator**

  Directed control flow connection.
  The object on the line side precedes the one on the arrow side in the control flow. E.g., an event triggers the function and a function results in the event.
  Functions and events can have only one incoming control flow arrow and one outgoing control flow connection.
  Join logical operators can have multiple incoming connections and only one outgoing connection.
  Split logical operators can have only one incoming connection and multiple outgoing connections.
- **Information Carrier – Function / Process Interface (both ways)**
  Directed connection.
  Information carrier on the line side is input to the function/process interface on the arrow side.
  Information carrier on the arrow side is output of the function/process interface on the line side.
  “1 to n“ connections are allowed in both directions.
- **Information Carrier – Application (both ways)**
  Directed connection.
  Information carrier on the line side is input to the application on the arrow side.
  Information carrier on the arrow side is output of the application on the line side.
  “1 to n“ connections are allowed in both directions.
- **Information Carrier – Organizational Elements/Cluster (both ways)**
  Directed connection.
  Information carrier/cluster on the line side is provided to the organizational element on the arrow side.
  Information carrier/cluster on the arrow side is provided by the organizational element on the line side.

"1 to n" connections are allowed in both directions.

- **Organizational Element – Function (both ways)**
  Undirected connection with no arrow, both directions having the same meaning.
  "1 to n" connections are allowed in both directions.
  The organizational element is responsible for the function with the role specified by the name of the connection.
  Possible connection names: *carries out, approves, supports, contributes to, must be informed on completion*.
- **Business Rule – Function (both ways)**
  Undirected connection with no arrow, both directions having the same meaning.
  "1 to n" connections are allowed in both directions.
  Business rule is applicable during execution of the connected function.
- **Business Rule – Reference (both ways)**
  Undirected connection with no arrow, both directions having the same meaning. Only defined in UPROM.
  "1 to n" connections are allowed in both directions.
  The reference is the source (reason) for the connected business rule.
- **Application – Function (both ways)**
  Undirected connection with no arrow, both directions having the same meaning.
  A function can be connected to only one application.
  The application supports the execution of the connected function.
- **Improvement – Function (both ways)**
  Undirected connection with no arrow, both directions having the same meaning.
  One improvement can be connected to one function.
  The function brings the connected improvement in the business process.
- **Risk – Function (both ways)**
  Undirected connection with no arrow, both directions having the same meaning.
  "1 to n" connections are allowed in both directions.
  The function is associated with the connected risks.
- **KPI – Function**
  Directed connection in one way.
  Any number of KPIs can be connected to a function.
  KPI is measured on execution of the connected function.
- **KPI – Event**
  Directed connection in one way.
  Any number of KPIs can be connected to an event.
  KPI is measured upon occurrence of the connected event.
- **Information Carrier – KPI**
  Directed connection in one way.

"1 to n" connections are allowed in both directions.
Connected information carrier is utilized to calculate the KPI.

- **Function – Technical Term**
  Undirected connection with no arrow, both directions having the same meaning.
  "1 to n" connections are allowed in both directions.
  The function description involves the connected technical term.

Possible sub-diagram assignments:

- **Function:** can have FTD, another EPC or FAD assigned as sub-diagram.
- **Process Interface:** can have FTD or another EPC assigned as sub-diagram.

Modeling Guidelines for EPC:

We utilize the following guidelines specific for EPC modeling. EPCs can become complex due to the control flow modeling and detailed structure of the models. These guidelines need to be followed to ensure systematic process analysis in EPCs and preserve them in a well-structured way.

- Restrict the Number of Nodes Utilized:

As specified by "7PMG", the more elements a process model has, the higher is the error probability [9]. Based on experimental studies, it is identified that frequency of errors in a model increases by more than 50% if the number of nodes exceeds a certain amount. Based on these studies, we aim to restrict the following types of diagram properties to the given numbers [175]:

  - $S_N > 48$ (number of nodes)
  - $S_c > 8$ (number of connectors)
  - $S_F > 40$ (number of functions)
  - $S_A > 40$ (number of arcs)

- Avoid OR logical operator if the model can be replaced with an understandable alternative

Usage of OR operators is suggested to be avoided in business process models [9]. However, keeping in mind that the analysis conducted in UPROM is in the business domain, it can be required to indicate in a readable way that not all or only one of the different paths need to be chosen. For example, if the sub-entities of an entity can be updated, just one, some or all of them can be updated. We need to indicate this need for both process flow and process execution if the process is to be automated. Still, OR operator can be replaced with alternative modeling, but this makes the models hard to read [77]. That's why UPROM doesn't guide users to totally avoid usage of OR connectors.

UPROM suggests the users to evaluate the alternative model to replace OR and use it if it is accepted to be understandable by users. An alternative to avoid usage of OR may be to merge all of the activities in between OR split-join pair to one function. In many cases, this will cause a very complex single activity which has a level of granularity which does not conform to the rest of the process.

Figure 13 illustrates how the OR operators can be avoided by an alternative modeling. On the left of the figure, one or any number of four functions can be executed. On the right, OR operator is replaced with And operator. Inside the "And" block for each alternative, XOR block indicates that a function can be executed if the given condition (event) holds, otherwise it is not executed.

Usage of OR connectors shall be avoided for multiple start and end events, as this causes ambiguity to decide when the process starts or ends. Additionally, as a rule in EPC modeling, XOR and OR operators as splits cannot be used after an event, because that event "cannot decide" which path will be chosen among the alternatives of XOR [173]. However, there are examples where this rule is not applied because of clarity considerations even by the definers of the notation [176].

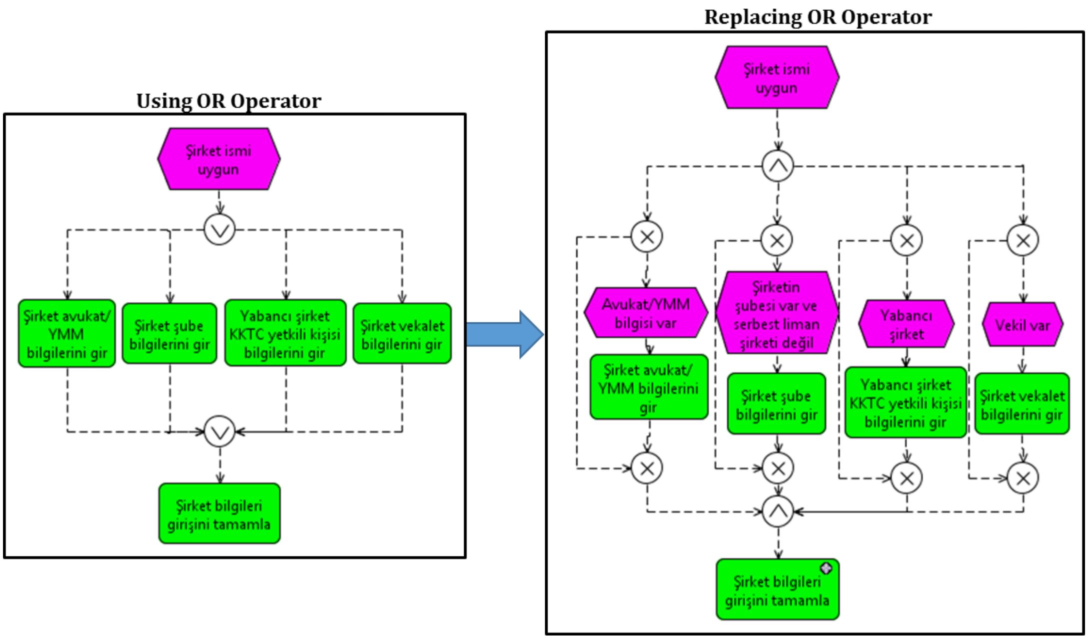


Figure 13 Replacing OR operator with alternative modeling

3.1.4. Diagram Type 4 – Organization Chart Diagram (OCD)

OCD is utilized to model the organizational perspective by placing all organizational elements in the modeling project and depicting the relations between them. It is also defined as a diagram type in ARIS framework [173]. OCD serves the same purpose with the traditional organizational charts. The difference is that, within the modeling project it covers all organizational element objects placed on EPCs, so it is assured that organizational structure is identified in relation with business processes.

The constructs are the same with the ones identified under "Organizational Elements" part of the previous section numbered 3.1.3.

Possible connections:

- **Organizational Unit – Organizational Unit**
  Directed connection.
  "1 to n" connections are allowed from line side to arrow side. "1 to 1" connection is allowed from arrow side to line side.
  Organizational unit on the line side is the superior or composed of the organizational unit on the arrow side as specified by the connection name.
  Possible connection names: *is superior, is technical superior to, is composed of*.
- **Organizational Unit – Position**
  Directed connection.
  "1 to n" connections are allowed from line side to arrow side. "1 to 1" connection is allowed from arrow side to line side.
  Organizational unit on the line side is composed of the position on the arrow side.
  Possible connection name: *is composed of*.
- **Position – Organizational Unit**
  Directed connection.
  "1 to n" connections are allowed from line side to arrow side. "1 to 1" connection is allowed from arrow side to line side.
  Position on the line side is the superior or manager for the organizational unit on the arrow side as specified by the connection name.
  Possible connection names: *is superior, is technical superior to, is manager for*.
- **Position – Position**
  Directed connection.
  "1 to n" connections are allowed in both ways.
  Position on the line side is superior or manager for the position on the arrow side, depending on the connection name.
  Possible connection names: *is technical superior to, is manager for.*

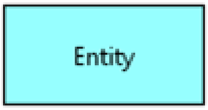
3.1.5. Diagram Type 5 – Function Allocation Diagram (FAD)

In ARIS framework, FAD is used to detail the relations specific to a single function on the EPC [173]. Thus, it is like a zoomed depiction of an individual function, so that EPCs remain less crowded and easy to read. In UPROM, we identify a different notation for FAD and utilize it to conduct function allocation analysis of the leaf functions to be automated in EPCs. FAD is utilized to identify responsibilities, entities utilized and manipulated, operations conducted on those entities and constraints to be considered for the execution of that function and software systems the entities reside on. As all analysis activities are conducted from the business perspective, these entities are conceptual illustrations of real-life concepts, rather than entities in a database model. The details of the analysis conducted for FAD modeling is described in section 3.2. Here, the constructs, connections and semantics for FAD are provided.

Organizational Elements: The collection of constructs representing organizational aspects of the processes as specified for EPC in section 3.1.3.

Function: The same construct utilized in EPC as described in section 3.1.3. The name of the function on FAD is assigned to be the same with the leaf function on EPC for which detailed analysis is conducted. Therefore, the two functions are assigned to be unique and in this way, the two diagrams are associated.

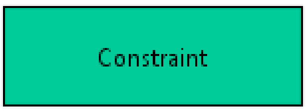
Entity: A representation of a real world entity in the business domain. Entities are identified as illustrations of real-life concepts, rather than well-defined entities in a database model, as the analysis is conducted from business perspective. Information carrier symbols in EPCs are utilized to identify entities in FADs. ***Additional attributes:*** link, technical term.



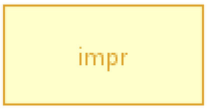
Cluster: It is a collection of data or entities as defined for EPC in section 3.1.3. It has the same semantics with the entity construct. ***Additional attributes:*** link, technical term.

Application: The same construct defined in EPC as described in section 3.1.3. There may be more than one application and entities connected to those applications on a FAD.

Constraint: It is used to specify the constraints applicable during the execution of the function on an application system. Business rules connected to the function on EPC are clues to identify the constraints. However, constraints are specific to the application system. ***Additional attributes:*** link.



Improvement: It is a construct added specifically for UPROM. It is used to specify the improvements achieved (or to be potentially achieved) by business process analysis.



Possible connections:

- **Organizational Element – Function**

  Undirected connection with no arrow, both directions having the same meaning.

  Many organizational elements can be connected to the function.

  Either all or any single organizational element is responsible for the function with the role specified by the name of the connection.

  Possible connection names: *carries out, approves, supports, contributes to, must be informed on completion*.

  If all organizational element – function connections on FAD have an additional label of "1" in their connection names, any of the organizational elements can conduct the related responsibility individually on the function, and the conduct of one responsibility of the same type is enough for the function to be completed.

  If there is no label of "1", all connected organizational elements need to conduct the related responsibility for the function to be completed.

  If there is no organizational element connected to the function on FAD, the related function is conducted automatically by the application system.

- **Function – Entity/Cluster**

  Directed connection available only one way.

  Many entities can be connected to the function.

  During and on completion of the function execution, the operations specified by the name of the connection are applied on the connected entities.

  Possible connection names: *create, change, delete, view, list, read, use*.

  If some function – entity tuples have an additional label of "1" in their connection names, it indicates that only one of those operations will be applicable depending on the situation. However, the system needs to have functionality to conduct all of those operations including the labeled ones.

  The semantics and rationale for assigning each of the connection names are as follows:

  - ***Create:*** It indicates the creation of an entity into a persistent storage of the application the entity is connected to. It is also used when data of an entity is sent to an external application.
  - ***Change:*** It is used when the entity already exists on the application and the attributes of the entity need to be changed.
  - ***Delete:*** When an instance of an entity needs to be deleted, this operation is used.

- ***View:*** This operation is used when the attributes of a specific entity, usually a previously selected one, are obtained and shown to the user. It indicates that the entity is needed to be displayed to the user so that the user performs the related function.
- ***List:*** It is used for situations where an entity is listed for all of its values, or it is queried by one or more of its attributes and the resulting limited set of entities are shown. Listing operation is also used when a listing (like drop-down list) is populated during the operations pertaining to other entities.
- ***Read:*** When it is required just to obtain the attributes of an entity, this operation is used. This happens together with other entity operations. Usage of read and list can sometimes be mistaken. The difference is that, read is used when a single entity needs to be retrieved and used in other operations, and doesn't need to be shown to the user. It is used if the existence of the entity is a prerequisite for the activity to be performed or the entity is input to another entity to be created or updated.
- ***Use:*** This operation is only used together with list or view to express the information utilized for those operations. For example, if a query is conducted on an entity, some attribute is utilized to make the search. That attribute is shown connected with a use operation so that in the requirements, we can understand the basis for the query.

- **Entity/Cluster – Application**

  Directed connection available only one way.

  Many entities can be connected to the application. An entity must be connected to an application and only one application.

  Entity/cluster is an entity of the connected application and the operations are conducted on that application.

- **Application – Constraint**

  Undirected connection, both directions having the same meaning.

  Many or no constraints can be connected to the application, a constraint must be connected to only one application.

  The constraint specifies the application, thus shall be considered for the connected application during the execution of the related function.

3.1.6. Diagram Type 6 – Entity Relationship Diagram (ERD)

A similar diagram exists in ARIS framework with the name eERM but different connection types between the constructs [173]. In UPROM, ERD includes all entities placed on FADs. On ERD, the relations between conceptual entities are defined by means of aggregation, generalization and generic relationship type connections. The conceptual data model depicted by FADs and ERDs in UPROM conform to the definition of "conceptual data modeling" in COSMIC Business Application Guideline: "*this level shows things in the real-world that are important for a piece of software and the relationships between them*" [158, p. 15]. In this way, an informal, business view model of the entities is formed. The stakeholders are ensured to have a complete view and understand the same concepts for the entities. The constructs, connections and semantics for ERD are provided below.

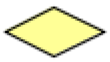
Entity: A representation of a real world entity in the business domain as defined for FAD in section 3.1.5. ***Additional attributes:*** link, technical term.

Cluster: It is a collection of data or entities as defined for EPC in section 3.1.3. It has the same semantics with the entity construct. ***Additional attributes:*** link, technical term.

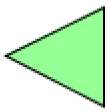
Attribute: It represents properties stored as a piece of data that characterize the entity. It is a part of the entity or cluster, used as connected to them. ***Additional attributes:*** technical term.

Key Attribute: It has the same characteristics with the attribute construct, only being a specific attribute to represent a property of the entity that provides uniqueness for it. ***Additional attributes:*** technical term.

Relationship: It is used to depict any type of relationship between entities/clusters with the name assigned by the user as the name of the construct.



Generalization: It is used to indicate that an entity is in higher level to the ones on the other side of the generalization construct, in the same meaning with superclass and subclass in object oriented design. When the generalization relation is read in the opposite direction, it indicates that lower level entities are specializations of the higher level one.



Possible connections:

- **Entity/Cluster – Generalization**
  Directed connection with no arrow.

Possible connection names: *is supertype of, no name (blank)*.
The connection with the name "is supertype of" shall be established only with one entity for the given generalization object and in the direction of "entity to generalization". This entity is high level with respect to other connected entities.
A generalization construct shall have one incoming connection and at least one outgoing connection.
The connection with no name shall be established with one or more entities for the given generalization object and in the direction of "generalization to entity". These entities are specializations of the high level entity.

- **Entity/Cluster – Attribute/Key Attribute**
  Undirected connection with no arrow, both directions having the same meaning.
  Entity/cluster can be connected to "0 to n" attributes and "0 to 1" key attribute. An attribute shall be connected to only one entity/cluster.
  Entity/cluster has the connected attributes.
- **Entity/Cluster – Relationship**
  Directed connection with no arrow.
  The connection in the direction of "entity to relationship" shall be established only with one entity for the given relationship object.
  The connection in the direction of "relationship to entity" shall be established only with "1 to n" entities for the given relationship object.
  A relationship construct shall have one incoming connection and at least one outgoing connection.
  The connection shall be read starting from the single entity connected in "entity to relationship" direction as: "entity" + "relationship name" + other entity names.
- **Entity/Cluster – Entity/Cluster**
  Directed connection with diamond shape.
  The entity in the direction of the "diamond" is the aggregate entity.
  The entities connected to the aggregate entity in the direction of straight line are parts of the aggregate entity. That is, aggregate entity is composed of the parts.

Possible sub-diagram assignments:

- **Entity/Cluster:** can have other ERD assigned as sub-diagram.

3.1.7. Meta-model for the Notation

Although the meta-model is critical for the definition of business process notations, publications on it are scarce [177]–[180]. Moreover, meta-model definitions usually include only one diagram type; and not the relation between notations of multiple diagram types. UPROM covers six diagram types for which object types are related across the diagrams. We need to define the meta-model for all diagram types as a whole and identify the relationship constraints for each diagram separately. The meta-model for UPROM notation is explained in two parts. The core meta-model covers the "inheritance" relationships between all constructs of the notation, shown in Figure 14. Behavioral meta-model, as shown in Figure 15, describes the relational constraints between the constructs. This meta-model identifies the possible connections and connection limitations between the constructs. On the UPROM tool, the connections between the constructs are limited to match this meta-model.

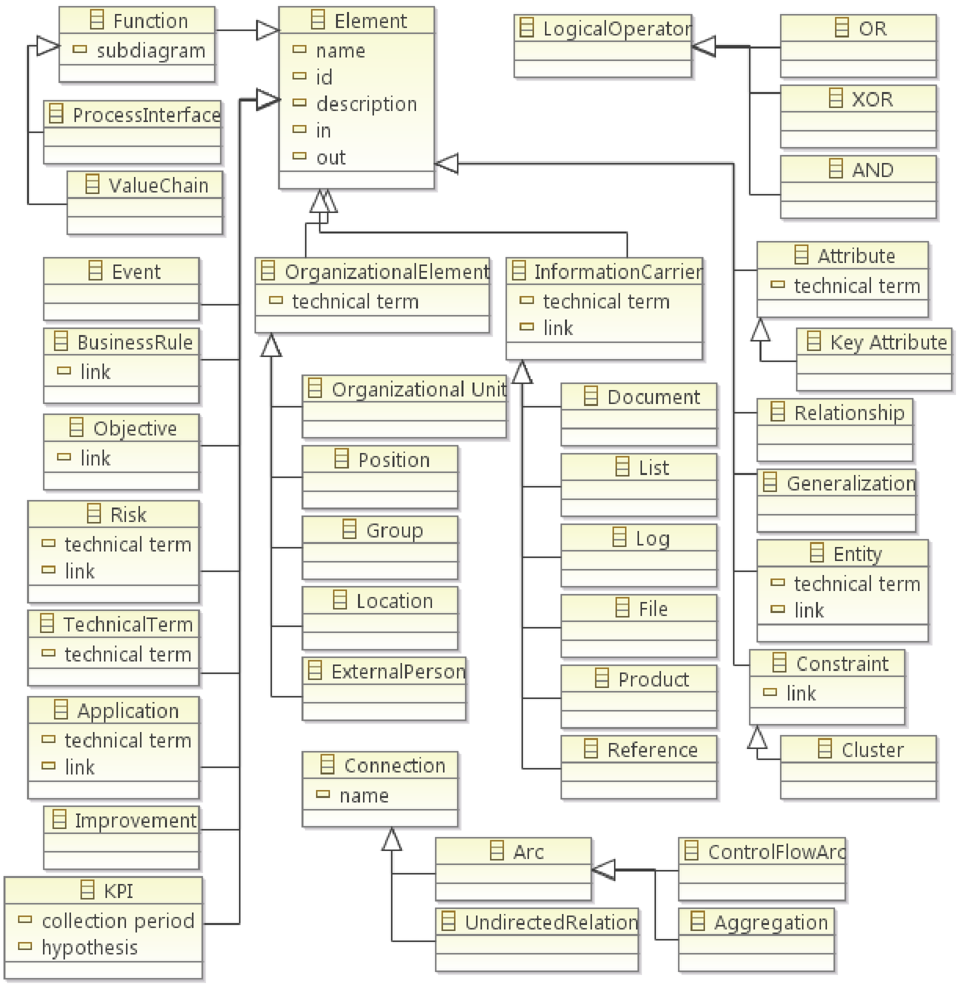


Figure 14 First part of meta-model for notation of UPROM

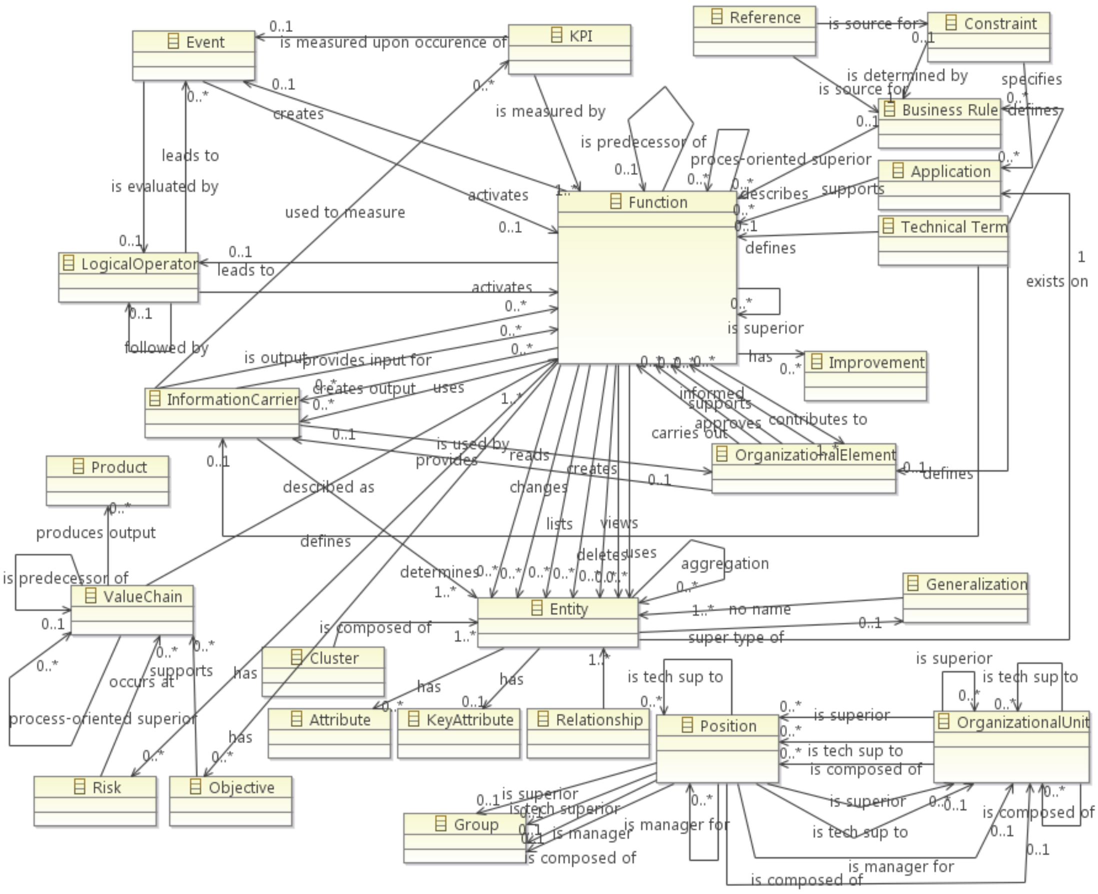


Figure 15 Second part of meta-model for notation of UPROM

3.2. The Process and Guidelines

A major problem in BPMod field is "how to model the processes" as it affects all the way down the improvement implementation. In most studies, while BPMod notation is explained in detail, the process of developing business process models remains unexplained or implicit. Current methodologies usually assume process modeling activities to be conducted in a top-down fashion [181]. That is, the overall processes are identified primarily and then a process improvement team captures the processes one by one by interviewing stakeholders of each process and then detailing those processes by resolving inconsistencies. This approach requires a complete understanding of the context and therefore can create the problems that process modeling takes months and the contribution of actual performers remains minimal; while requiring the least amount of BPMod expertise in the organization. Top-down approaches may also be implemented in an iterative way. To speed up the studies, parallel groups of BPMod experts may be put into action. Distributed, bottom-up approaches may take longer times to implement, but it creates BPMod expertise local to the organization. The BPMod process to be applied shall be explicitly determined and chosen depending on the structure of the organization, processes to be analyzed and the goals.

There are some studies in which modeling process is examined, but these handle only high level phases and key aspects [165], [182]–[184]. Here, we aim to define the modeling process specific to UPROM methodology. Activities for business process improvement are not included in this process as it is a wide field of study on its own [4]. Additionally, detailed activities to develop models conforming to the notation are not described here, as details are provided in sections 3.1 and 3.2.2. The process of UPROM is composed of three main activities as shown in Figure 1. These activities are:

- Descriptive business process analysis
- Function Allocation Analysis
- Automated Artifact Generation

Sub-processes for the first two activities are described below in sections 3.2.1 and 3.2.3. Guidelines to be followed during these activities are defined in section 3.2.2 for the first activity and in section 3.2.4 for the second activity. The details of automated artifact generation activity are provided in section 3.3.

3.2.1. The Process for Descriptive Business Process Analysis

The process to be followed in the first step of UPROM process is illustrated in Figure 16 using EPC notation. Only the control flow is modeled and details like inputs and outputs are omitted from the diagram for the sake of simplicity. Details of each function in the process are described below. We define the overall output of this step as "descriptive business process models".

Identify scope of BPMod studies:

The scope that will be covered by BPMod studies in the organization is defined in this initial step. The scope may be limited to a specific organizational unit, or may cover many or all of the functional processes in the organization. BPMod studies must be conducted in alignment with strategic objectives. Thus, organizational vision, strategic plans, legislative documents are important inputs to identify the scope. As an output of this activity, high level processes that will be included in the studies are designated.

Identify goals for BPMod:

The goals to be achieved by means of BPMod activities need to be identified. The decisions on the scope from the previous step specifies the goals. The artifacts the organization requires as outputs of BPMod are specified. The rest of the activities are conducted according to this decision. For example, if process automation is not a goal, the second high level activity, "function allocation analysis", is not carried out.

Plan BPMod studies:

The roles and responsibilities for BPMod are determined. The main roles are analysts (modelers), domain experts and sponsor. Other roles that can be needed during BPMod studies and their description can be found in [183]. Resource needs are identified, resources for each role are defined and duration, effort and teams are planned. As the output of this activity, a plan similar to a project plan including the activities for the rest of the BPMod studies with resources, dependencies and schedule is developed.

Define process map diagram:

The highest level decomposition of processes is identified as process modules and modeled in a process map which is the highest level functional diagram in the modeling project. A process map can be VC, EPC or FTD diagram type. The detailed information regarding the process map is provided in section 3.2.2.

Design hierarchical structure:

After the first level of decomposition by means of process map, the processes are decomposed further into lower level process modules. The hierarchical folder structure is established compatible with the process decomposition. Identification of process modules and establishment of the folder structure are described in section 3.2.2. As the processes are decomposed into a modular and hierarchical structure, the BPMod work can be easily distributed to agile sprints or iterative life cycle iterations, or conduct at one time by sharing the work among the members of an analyst team in a waterfall model. Process modules are input for the subsequent planning activity.

Assign teams and plan workshops for each process module:

As processes are decomposed as coherent process modules, BPMod activities can be planned separately for those modules. Each process module needs to have its own team assigned and a plan to conduct workshops with domain experts and analysts. The plans may be conducted for only the highest level process modules, or further specified for lower levels, too. This depends on the extent of the scope and nature of process modules.

Develop business process models of type VC, FTD, EPC and OCD:

This is the main activity to develop the models conforming to the notation and guidelines defined in section 3.1 and 3.2.2. The modelers depict the decomposition of processes by using VC and FTDs, and identify detailed control flow by means of EPCs. During the modeling, an OC diagram is created for the modeling project. All organizational elements having a role in EPCs are placed on this OCD and their relations are modeled. Process information is elicited and analyzed through workshops. Additionally, available process documents, legislations and laws applicable to the organizational domain are also important sources of information.

As described in previous paragraphs, this activity can be conducted in parallel for all process modules by different teams, in an iterative way, or all process modules can be analyzed by one team in the order defined in the plan. Depending on the lifecycle applied and the plan, when the step of descriptive business process analysis is completed for one process module, the team can continue on to the second step, function allocation analysis for that module. As-is and to-be models can be developed separately; or after completion of as-is model development, to-be models can be further specified on the same models.

Depending on the goals identified for BPMod, the following activities are conducted in parallel to modeling, to further detail the descriptive process models.

- ***Add process meta-data:***
  General information regarding a process named as "process meta-data" are added to every model of type EPC, FTD and VCD. This property enables users to describe the identity of the processes, record its key information and provide a detailed description in natural language. This type of information is usually provided in natural language process documents. The following process meta-data is identified:
    - Model name
    - Purpose
    - Scope
    - Status
    - Version
    - Description
- ***Add detailed descriptions for objects***
  Detailed natural language description can be added to every object in UPROM. But in practice, we observe that users especially require to describe functions in detail. Even if all natural language descriptions can be reflected in the models, in such a modeling style the abstraction level may become too detailed and complexity of the model exceeds the readability limits. To limit the complexity, the details of the functions can be preferred to be described in natural language. Adding function descriptions on the models are especially important, as they are later utilized during the generation of process definition documents.
- ***Add terminology descriptions and links:***
  As defined in the notation, all constructs that have an additional attribute of "technical term" can be assigned a terminology definition. These constructs are of information carrier, organizational element and entity types, which are usually associated with a terminology in the business domain. If there is additional need for defining terms other than existing constructs, technical term construct is utilized to further define terminologies. By means of the uniqueness of objects, it is assured that an object has only one definition through modeling project. Technical term attribute can be assigned for the following constructs:
  "product, risk, application, organizational elements, information carriers, entity, cluster, attribute, key attribute, technical term".

In addition to terminology definitions, links are assigned for specific types of constructs when needed. The links are used to reference to external documents that describes and provides more information on the related object. It is important to store related documents in the repository within the folder structure of the modeling project. The following types of constructs can be assigned a link attribute:
"objective, product, risk, business rule, application, information carriers, entity, cluster, constraint, attribute, key attribute".

- ***Identify improvements***
  The improvements introduced related to functions and general to the processes are defined using the improvement construct. Each improvement is described in a separate object, and detailed description of the improvement is given in the description attribute. An improvement can be related to a function by connection, or can be general to a process and placed as an unconnected object on the diagram.
- ***Define process KPIs***
  Key performance indicators, or metrics regarding the process are defined on EPCs. The details of a KPI object must be identified to ensure that analysts can grasp how it will be exactly measured in process automation. Different aspects of KPI construct and how it is utilized to describe the KPI are as follows:
  - Connect KPI to function if KPI is to be measured during execution of that function.
  - Connect KPI to event if KPI is to be measured on the occurrence of that event.
  - Connect KPI to information carrier(s). These information carriers provide the source of information by means of which KPI can be calculated.
  - Identify collection period and time as the attribute of KPI.
  - Identify hypothesis of the KPI as the attribute. This is used to analyze the results of metrics collection and identify when the resulting values are beyond acceptable values.

Validate Descriptive Business Process Models:

After the completion of descriptive business process models, to finalize this stage, end users and domain experts need to validate the models to ensure that they accurately reflect the organization. This can be conducted by delivering the models to the experts, or by conducting workshops and presenting them the outputs. This activity can be conducted in a single step for the whole study, or separately for each process module.

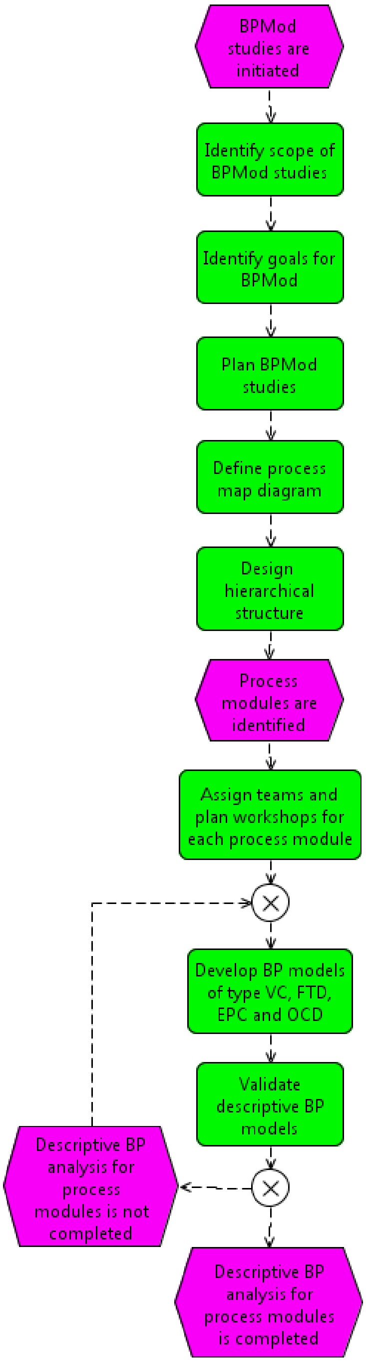


Figure 16 Descriptive business process analysis process

3.2.2. Guidelines for Descriptive Business Process Analysis

Level of granularity:

Level of granularity must be uniform across the requirements [125, p. 28], the same need holds for business process models in our case. It is necessary to keep the same level of granularity through the modeling project in UPROM. If a process goes down to a detail where it defines a separate function to update each of its detailed attributes, where another process in the same project places only one function for updating an entity with many sub-entities, we can say that the level of granularity would not be the same. Sometimes, this difference is natural in the processes, as different types of activities need different detail levels. For example, while the user needs separate functions to enter different sub-entity information regarding to a main entity, she may need just one function to view the whole entity, as she requires viewing all information. In these cases, UPROM suggests to compensate for this difference on FADs, by placing all related entities on the diagram. In this way, although the activity seems to be less detailed, the analysis on FAD covers the necessary detail and the requirements and size measurement are generated for the same detail level.

Modeling Project and Predefined Structure

Decomposing concepts into multiple levels and depicting the information in hierarchical forms is a common way to decrease complexity [122]. The decomposition of process modules is utilized in UPROM to organize the structure of the modeling project. In this way, process modules are clearly defined and become easy to understand in the modeling project; while also artifact generation mechanisms can be easily executed.

Within a modeling project, all functional and behavioral diagrams, including VC, EPC and FTD, must be connected to each other by sub-diagram relationship. The highest level VC, EPC or FTD diagram in the modeling project is called the process map. It represents the decomposition of highest level process modules in the modeling project. None of the diagrams of type VC, EPC and FTD can exist independent from the hierarchical structure.

The concept of "process module" is used for a group of process models which are coherent and focus on a specific working area of the organization (like budgeting). Each process model encapsulated in a folder at any level can be treated as a process module which can be called by another process like a function call in a software code. All of the process models under a process module are connected to the hierarchical structure of the related module. Process modules can include lower level process modules inside it. The structure of the modeling project (including different diagram types, folder and sub-diagram structure) must conform to the following rules.

- There shall be a VC, EPC or FTD diagram file in the top level which references to one-level low diagrams in the hierarchy of the modeling project. This file is called the "process map diagram" and will be only one.
- There shall be no FAD diagrams in the top level hierarchy of the modeling project.
- Each sub-diagram of type VC, EPC or FTD shall be placed under a folder with the same location with the higher level diagram this sub-diagram is referenced to. The name of the folder shall include the name of the sub-diagram file.
- The rules for sub-diagrams will not be applied to the ones referenced from process interface symbols. The sub-diagram referenced from the process

interface shall also be referenced from a function or value chain symbol following the rules explained above.

- FAD diagrams shall always be referenced as a sub-diagram from an EPC or FTD diagram and never exist alone independent of the hierarchy. FAD diagrams shall be placed under the same folder with the EPC or FAD diagram from which it is referenced.
- OCD and ERD can be placed at any place in the folder structure. As these models will be only one or a few for the whole modeling project, it is preferable to place them at a high level together with the process map.

An example modeling project structure conforming to these rules can be seen in Figure 17. FADs are hidden in this project structure for the sake of obtaining a reasonable size. They are placed together with EPC files they are referenced from. It can be observed that no additional folders are created for FADs.

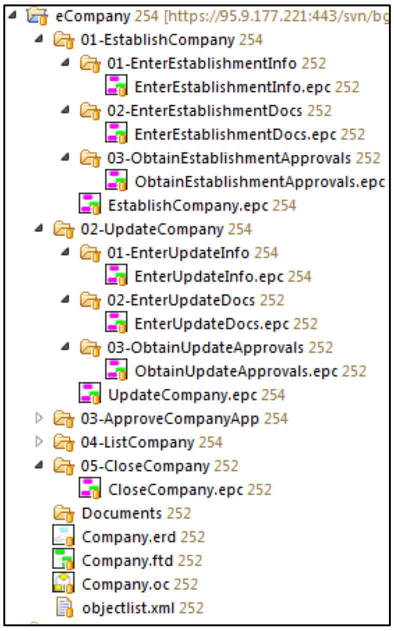


Figure 17 An example modeling project structure (FADs hidden)

Multiple Start and End Events and Uniqueness of Events:

Multiple start and end events are not encouraged as they increase the error probability in processes [9]. However, they are useful to enhance the expressiveness of the models in the business domain, and can result in less number of total control flow nodes in a modeling project; as some of the activities are grouped for more than one start and end event. By decreasing the number of nodes which are potentially similar activities, the error probability also decreases and the maintainability increases as only one group of nodes will be required to be updated when the process triggered or ending with those multiple start and end events change. Special focus shall be given to match start and end events so that flow is smooth between the diagram, sub-diagrams and process interfaces.

Models are ensured to be structured by means of using splits and joins in pairs as a block structure [185]. However, we need to keep in mind that in these two cases, we cannot use splits and joins in pairs:

- For multiple start and end events
- For feedback returns; which are used to indicate some part of the process to be restarted because of a feedback in process activities

In UPROM, in correlation with the notation described by IDS Scheer [50], multiple instances of events are utilized to provide a conceptual flow between the pieces that compose a modeling project. Thus, if an event is used to denote the end of Process 1, and the same event is used as a start event of Process 2; we understand that these two processes follow each other and the end of Process 1 triggers the start of Process 2. The start and end events before the trigger of a sub-process or a process interface must match with the detailed process of the sub-process or referenced process module. The usage of the same start and end events are illustrated in Figure 18. The events with the same name are unique. This enables event analysis across process models. Users can follow relations between different processes, analyze and identify dependencies based on events. Also, when the name of an event in one of the models is updated, all other related ones are also updated. In this way, consistency among the processes is ensured.

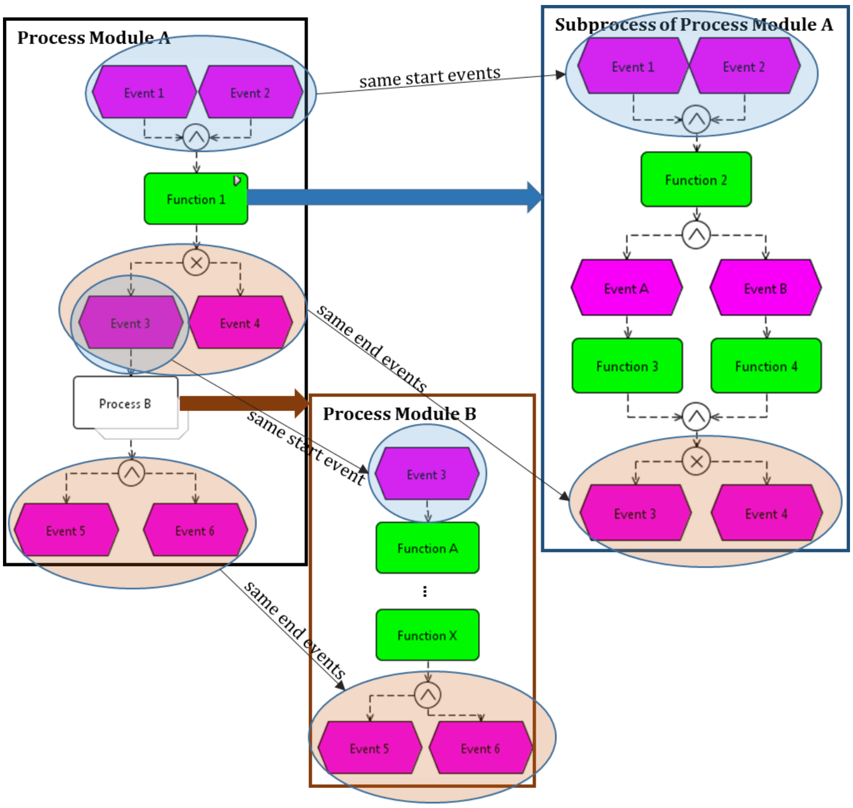


Figure 18 Illustration for using the unique start and end events between processes

Unique Objects through the Modeling Project

Objects of the same construct and name are maintained as unique objects in UPROM. The uniqueness enhances analysis activities. Examples of these are the analysis of start and end events as described above, identification of all occurrences of specific information carriers and so on. Altering a property of one of the instances of a unique object results in the same change in all instances of the unique object.

In addition to the same constructs, objects from the same group of constructs which are assigned the same name are maintained as a unique object. For example "order" document in one EPC, "order" file in another EPC and "order" entity in an ERD are all the same objects. The groups are identified so that logically related constructs are assigned to be unique. For example, any object of type information carrier and entity are unique if they have the same name, as the entities in FADs and ERDs are utilized to represent information carrier objects of EPCs. The objects of the following groups can be assigned as unique:

- Information carriers, entity, cluster, attribute, key attribute, technical term
- Function, process interface, value chain

Usage of Process Interface Construct in EPCs

Definition of reusable business processes during BPMod is a critical property to organize a hierarchical structure and increase understandability, decrease complexity and enhance maintainability. This is a need in different notations. For example, in URN, definition of modular process models is provided by stubs and plug-ins [141].

According to the basic EPC notation described by IDS Scheer, process interfaces are used to break down a process into smaller sub-processes; or to provide a reference to the initial process that is executed before the current process starts or the following process that will be executed after the current process is finished [97], [173]. In this idea, process interface construct is used to ensure the following processes to be visible, with the same idea as matching start events and end events, as described previously. The symbol appears only at the beginning or the end of an EPC diagram, and make it a visual reminder of the previous and following events in the whole picture. To summarize, in ARIS notation, process interfaces are used to:

- Represent upstream or downstream processes
- Provide an easy way to navigate through these processes

According to UPROM, and as supported by many field experts [174], the meaning brought by this interpretation is already provided by the use of multiple instances of events, even in a more flexible way. In UPROM, we utilize process interface construct as a way to provide a reference to an external process that is utilized by the current process. In this way, "reusable" processes are defined, which can be called like a black box module from any process. Many processes that call the same type of processes (usually, support processes like sales, configuration management etc. are required to be used in this way) need not know the details of the support process but just “call” it to execute like calling a function in a software code, and wait for the result.

There can be two situations for using the process interface as a reference:

- The process referenced by the process interface symbol is initiated, and the main process flow does not wait for this external process to finalize. In this case, process interface is shown with a connector and no flow occurs after the symbol.
- The process referenced by the process interface symbol is initiated, and the main process waits for the external process to go on. In this case, the process interface object is trapped between the symbols of the process, and the flow continues after the process interface.

As long as events are used properly, it is advised to use start and end events of the EPC model which is referenced by process interface symbol. There may be only one event shown on the process that uses the support process before and after the process interface object. But the details of the support process need to encompass all the events utilized by different processes that use the support process. Thus, for example, if a process module is called from “n” different processes, and it has different triggering entity in each of them, EPC model of that process module needs to have “n” triggering events. This usage is illustrated in Figure 19.

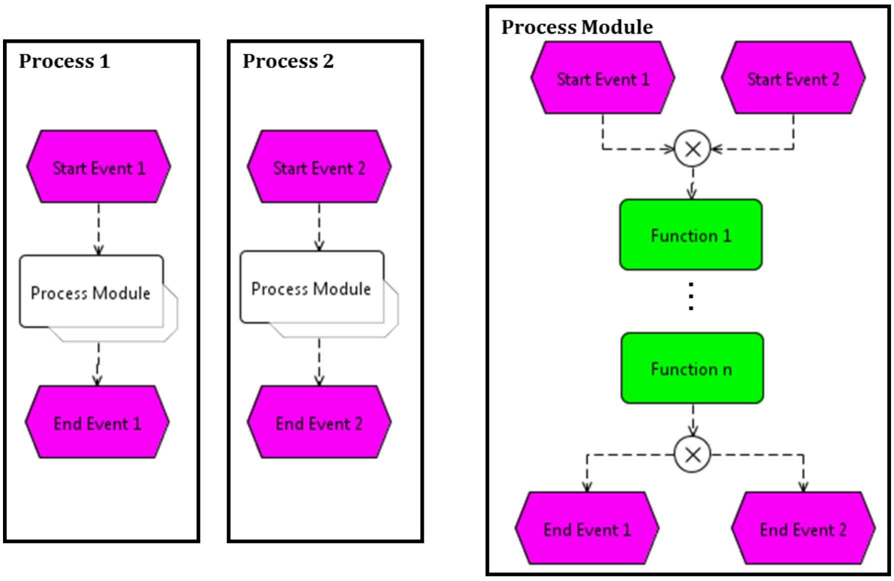


Figure 19 Illustration for usage of process interface construct

This type of usage for process interface provides the following advantages:

- Unrelated processes do not need to learn the internal details of the external (support) processes,
- Support processes are defined modularly, there is no duplication, thus the maintainability is higher,
- If the support process changes, the other ones using that process do not need to change as long as the interfaces between them do not change.

Usage of Business Rules:

Business rules usually emerge because of policies and rules applicable for the organizations. It has been revealed that BPMod users mostly need representing business rules in their models [91]. To fulfill this need, alternatives like narrative descriptions, spreadsheets, external tables, attributes or additional tools with hyperlinks can be used. When the aim is analyzing business processes from a technological perspective and developing such workarounds may not be satisfactory. However, from the business point of view, it is important to place information on business rules in an understandable way and in relation to the function for which it is applicable.

As classified by Goedertier and Vanthienen, the following business concerns can constitute business rules for business processes [186]:

- *"Business regulations: external directives: laws and contracts.*
- *Business policies: internal directives; strategies and procedures.*
- *Costs and benefits: the incurred benefits and costs of an activity.*
- *Time: the overall time to process an activity.*
- *Information prerequisites: the information required to start an activity.*
- *Technical and common-sense constraints"*

Another study groups business rules as operative rules which are enforced because of policy and structural rules to determine when something is true or not according to the category of the related case [27]. Goedertier and Vanthienen indicate that different categories of business rules are usually taken into consideration implicitly while developing process models. Some of these concerns are naturally included in the business processes by developing control flows, conditions and information flows (e.g. alternative paths, parallel flows). But for some of the business rules, it is not possible or practical to embed them inside control flow or information flow (e.g. time constraints). Though there are methods that provide formal ways of describing business rules in process models [123], [186], [187], as we develop models to support analysis of business processes by means of enhancing human understandability, we do not find it practical to implement these formal ways. However, we guide the users to keep the categories for business rules in mind during modeling so that related business rules can be revealed.

We specified guidelines especially important or specific to UPROM in this section and in section 3.1. In addition to these guidelines, we suggest that modelers consider following other guidelines applicable for BPMod activities especially for control flow modeling in the literature [9], [185], [188], [189]. Most of these guidelines focus on the visual properties of models rather than semantics on using notational elements. The specification of the visual properties of models is left out of scope of this study.

3.2.3. The Process for Function Allocation Analysis

The process to be followed in the second step of UPROM process is illustrated in Figure 20. This process model is more detailed than the previous one, as the activities of developing FAD and ERD are specific to UPROM. Details of each function in the process are described below. We define the overall output of this step as "complete set of business process models". Roles and responsibilities identified and the plan defined in the previous step is applicable for this step, too. The activities in this step are also conducted through workshops as in the previous step where applicable.

Identify and Design Applications for Process Automation:

Some studies in the literature state that explicit representation of software components and software interfaces are required in process models [123]. In process models, software systems that will be developed for process automation will be represented by application construct. In this activity, modelers shall decide if process automation software will be represented as one single application, or if it will be broken down into higher software modules as part of a high level software architecture design. If it is possible to identify software modules, it is recommended to design and utilize them on the models. In this way, a more detailed analysis of each software module can be achieved. Architectural design decisions shall be recorded somewhere to rationalize the applications placed on the models.

In addition to internal applications placed on FADs, frequently external applications from which data is obtained or to which data is sent, are also placed on FADs. Such applications are also identified in this step.

Identify Functions Allocated for Automation:

EPCs within descriptive models are examined one by one. Each leaf function in EPCs is analyzed to identify if that function will be conducted in an automated way in the software. For the allocated functions, FAD is created and assigned as a sub-diagram. The function in the EPC is placed in the center of this FAD. Thus, both functions are maintained as unique objects. In this way, EPCs and FADs remain totally aligned, as they are related by means of unique function objects.

Identify Entities Required and Manipulated by the Function:

Starting from this step, analysis activities are conducted for individual FADs created for a specific function. Analysts evaluate which entities will be required and manipulated during the execution of the function in process automation software. Information carriers connected to the function in EPC can be used as a clue to identify the entities. If they are defined to be the same, they will be associated as unique objects. An information carrier may be defined as an entity or decomposed into smaller entities. EPC, they can be already in digital form or they can be digitized as entities on FADs.

Identify Responsibilities to Execute the Function:

The responsibilities to execute the function in FAD are identified. Organizational elements are already connected to the function in EPC. However, the expected responsibilities may change when the function is automated, thus different roles can be assigned in EPC and FAD. Furthermore, in EPC general roles may be identified for the function like organizational units. In FADs, such responsibilities shall be refined into specific types if possible. For example, rather than organizational units, positions shall be preferred.

The responsibility type can be assigned as one of the alternatives described in section 3.1.5: *carries out, approves, supports, contributes to, must be informed on completion*. These are well-known and applied types used to identify the responsibilities of different tasks in an organization and in software systems [26], [27, p. 29].

If there is only one organizational element connected to function, this connection can only be named as "carries out". One or more responsibility may exist for the function at the same time. Multiple roles can execute the function with the same responsibility (e.g. two or more roles together can carry out the function), or with different responsibilities (e.g. one role carries out and one role approves). Sometimes, it is the situation that one role *or* the other can execute the function with the identified responsibility. In this case, the connections are labeled with "1".

Depending on the decomposition of functions and process automation decisions, sometimes, a manual function in EPC can become completely automated; thus no organizational element needs to be connected to the function. In such a case, a function not connected to an organizational element means that the function is conducted in a fully automated way by the related application on FAD.

Define Operations on Entities:

For the entities identified before, the operations to be conducted on each entity during or upon completion of function execution are identified. Operation types are:

- *create, change, delete, view, list, read, use*

The list of operations is inspired from the well-known CRUDL transactions which define the life cycle of a piece of data [158] and extended to meet the needs of different cases for requirements and software size estimation. The rationale for using each type of operation is described in detail in section 3.1.5. Operations on each entity for the function are identified based on the described rationale. More than one operation can be conducted on an entity, which are modeled as separate operations. Each entity is also connected to the application on which it resides.

Define constraints to automate the function:

The constraints which affect how the function will be executed in process automation software are identified. The business rules connected to the function on EPC are clues to identify the constraints. However, constraints are only applicable for the application system and restrict how the system will run. The constraints are identified as a full sentence that indicates how the system shall behave, so that the expression can be utilized directly to generate a requirements sentence.

Place the Entities on ERD:

In parallel to FAD modeling, at any time when new entities are identified on FADs, those entities are placed on ERDs, too. Then, relations between entities of a specific application including generalization, aggregation and generic relationships, are defined based on the principles defined in section 3.1.6 and guidelines provided in the following section.

Model Responsibilities, Entities, Systems and Constraints on FAD:

As defined in previous steps, all constructs are organized on a FAD. An example FAD can be seen in Figure 21.

Evaluate and Update Terminology Definitions:

The terminology of the domain as used in business processes were already defined in the previous step during descriptive business process analysis. However, as entities are defined in this step, new definitions may be needed. The modelers shall revise the models, identify if new terminology is needed to be defined, and add the definitions if necessary. It should be considered that if entity are information carrier objects are unique, terminology defined for the information carrier will be applicable for the entity.

Validate Complete Set of Models:

After the completion of this step, to finalize this stage, end users and domain experts need to validate the complete set of models. This validation can also be conducted by delivering the models to the experts, or by conducting workshops and presenting them the outputs.

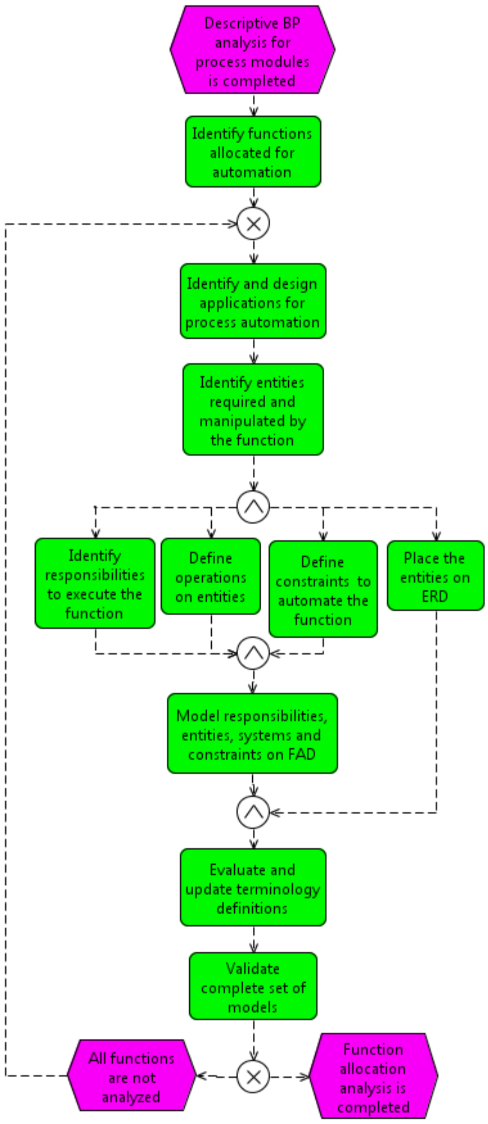


Figure 20 Function Allocation Analysis Process

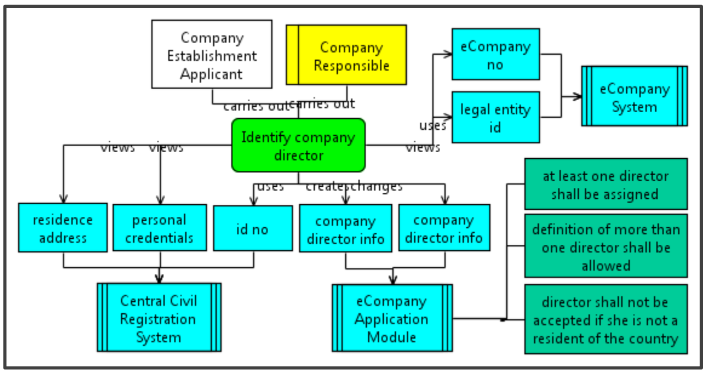


Figure 21 An example FAD

3.2.4. Guidelines for Function Allocation Analysis

Modeling Applications in FADs

"Application" placed on FAD models the systems on which the connected entities reside. These applications are mainly software systems that are planned to be developed to automate the processes. There may be more than one application in FAD. But one of them is the main application, the other systems are utilized to get information from or to send information to. To identify one main application on which the function is conducted, the application which is connected to the related function on EPC is taken into account.

Using Entity and Constraint on FAD to Identify State Changes

The change of states for the entities through the activities of a process is critical to identify its behavior at runtime [144]. A state of an entity refers to a number of attributes with certain values. At business analysis level in UPROM, details on changes of each attribute are not analyzed. However, for the critical entities identified, it is suggested to define a "state" attribute (or entity) as part of that entity and model the "change" of state entity on the related FADs. The description of how the state is changed is provided in "Constraint" object, and this is also considered as a requirement of the system. So, one can follow when the states of certain entities change by finding the state entity through the modeling project.

Structuring of ERDs

In traditional ER models, data is broken down into small chunks as entities and their relations are depicted in low level. On process based systems or in systems like document management systems, an entity meaningful to the organization in business process level is represented, which can be a cluster of different kinds of data [190]. In UPROM, the entities on ERDs are modeled in this style. ERD diagram type is utilized to compose a holistic view of the entities in the system and their relationship between each other. On ERDs, it is not aimed to provide all entities and attributes regarding to the system. Rather, the information that is required during the process flow is modeled. Moreover, ERDs are not an attempt to provide a fully structured ER model on which the software development can be based. This is rather a conceptual schema of what entities need to be carried and processed during the implementation of process models. Thus, the current situations may arise during ERD modeling:

- Not all of the information is placed on ERDs. If there is an entity called "Company", all the information and related entities regarding to "Company" are not necessarily shown on the ERD.
- The chunk of information for an entity which would rather be shown as attributes may be represented as entities on ERDs, so that operations on that chunk of information can be modeled on FAD. Moreover, an attribute on ERD can be represented as an entity on FAD as long as they have the same name. E.g.

company entity may have an entity connected to it with aggregation named as "Company Contact Information". This entity is composed of many attributes including address, phone numbers, e-mails etc. as modeled on the ERD. On FAD, an entity is placed for related attributes and in this way, specific operations conducted on this chunk of information during process flows are depicted. We need to keep in mind that to express the group of information as an entity, the operations shall be valid for the whole group. It is a good practice to detail descriptions of these entities and define attributes on a separate documentation.

- On a traditional ER model, relationships with M-N connections with entities are treated as separate entities, as they need to have their own tables on the database. For the sake of simplicity, in UPROM, this kind of relationship-entities are not shown, as it is not possible to conduct operations on relationships on FADs. If there is such a need, relationship-entities shall be modeled as entities.

3.3. Automated Generation of Artifacts

After development of models through the first and second steps of UPROM process, users can now automatically generate the artifacts to be used for other purposes. Artifact generation is conducted depending on the scope of BPMod studies identified during the first activity. If the organization does not have a goal for process automation, it doesn't need to conduct step 2: function allocation analysis. Then related artifacts for process automation also cannot be generated. The users can generate process documentation and process metrics list at the end of step 1: descriptive business process analysis. User requirements document and estimated software size can only be generated after the completion of step 2. As mentioned before, UPROM can be followed in any life cycle. UPROM artifacts used as inputs in an iterative life cycle are visualized as an example in Figure 22. Generation procedures for each of the artifacts are provided in the following sections. Note that, when previous steps are reapplied and an update is made on the models, this step shall be conducted again to easily obtain the updated artifacts.

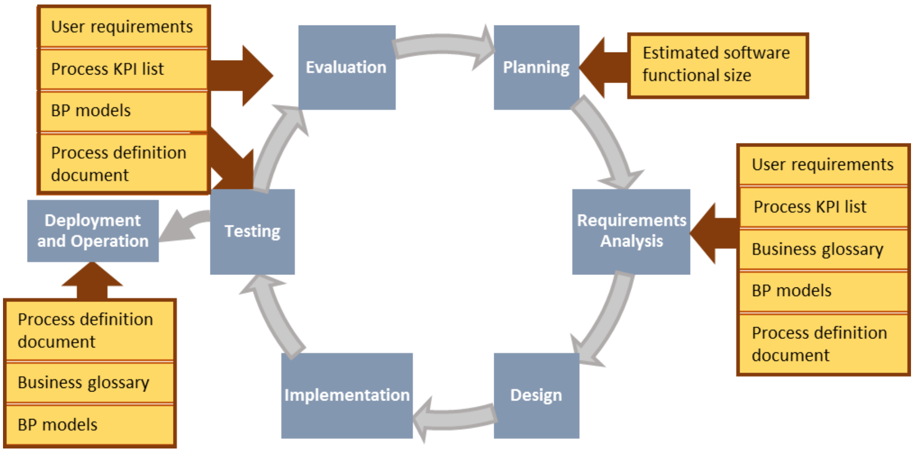


Figure 22 UPROM artifacts used as inputs in SDLC

In UPROM, all artifacts are generated in Turkish. For requirements generation, as described in the following section, the sentence structure must be rebuilt for the English language. However, the rest of the artifacts can easily be prepared in English when the names in the template are replaced with English versions on the tool. The definition of templates for the artifacts are provided in English in this chapter. However, the same fields are generated in Turkish by the tool, as will be seen in the following chapter.

3.3.1. Generation of User Requirements Document

Identification of Requirements Statements:

Requirements statements are identified by using FADs. The sentences are formed in Turkish language. Three types of requirements are identified. The procedures for generating the sentences are described below. Bold words in sentence templates indicate the name of the objects on FADs.

Type-1 Requirements Sentence:

Type-1 requirements sentence specifies the responsibilities on the related function. The structure in natural language is in the following form:

"**activity**(me/ma) işlemini **role1** yürütecek, **role2** onaylayacak, **role3** destekleyecek, **role4** katkı verecek, **role5** bittiğinde haberdar edecektir."

- **activity**: Name of function symbol in FAD diagram
- **role**: The name of the organizational element in FAD. There may be three cases:
    - In every case, the sentence starts with the following statement:
      "**activity**(me/ma) işlemini"
      Considering that all of the function labels are given in imperative form in Turkish (e.g. işi yap), to use this statement in sentence, a suffix of me/ma is added at the end of the function label. The type of the suffix (me/ma) is decided according to the last vowel letter of the word. If it is a high vowel, "ma" is added, otherwise "me" is added.
    - If there is only one organizational element connected to function named as P, this connection can only be named as "carries out". Then, only P is placed for "**role**" and the rest of the sentence is finished appending the following statement:
      "**role1** yürütecektir."
    - If there are n organizational elements connected to the function named as "P1, P2, …,Pn", and these roles are all connected with the same connection type, the statement "P1, P2, … ve Pn" is used for "**role**" and the related Turkish statement is appended for the connection type as explained in Table 3. The part of the sentence is in the following form:
      "**P1, P2, …ve Pn** yürütecek/onaylayacak/bittiğinde haberdar edilecek/destekleyecek/katkı verecek"

- If the case is the same as the above but the connections of the same type are labeled with a number (e.g. like carries out 1, approves 1), this indicates that "one or the other organizational element can conduct the related activity. In this case, the related part of the sentence is formed as: "**P1, P2, ...veya Pn** yürütecek/onaylayacak/bittiğinde haberdar edilecek/destekleyecek/katkı verecek"
  Note: For a given FAD and the operations on one application, if there is one type of organizational element – function connection (e.g. carries out), that connection type must be either numbered or not. Both numbered and not-numbered connection names cannot be used.
- If there are more than one connection types on FAD, organizational elements are grouped according to the types and sentence parts as stated above are appended in the order given in Table 3 and separated with semicolons.
- The sentence is completed by appending the verb suffix (as provided in Table 3) appropriate for the last connection type added to the sentence.
- If there is no organizational element connected to the function, it indicates that the function is conducted automatically by the system. In this case, the sentence is completed with the phrase: "sistem otomatik olarak gerçekleştirecektir".

Table 3 Turkish statements corresponding to different organizational element - function connection names on FAD

Organizational Element – Function Connection Name	Turkish Statement	Verb suffix
carries out	yürütecek	-tir
approves	onaylayacak	-tır
must be informed on completion	bittiğinde haberdar edilecek	-tir
supports	desteklenecek	-tir
contributes to	katkı verecek	-tir

Type-2 Requirements Sentence:

Type-2 requirements sentence is used to express the operations conducted on the entities on FAD. By means of these operations, we understand the entities that are utilized, created, changed, written or deleted during the related activity. The structure of the sentence in natural language is in the following form:

"Bu işlem sırasında **Application** sisteminde/n **operations on system**".

- The name of the connection between Entity and Application objects in Turkish is used for "**operations on system**" statement. Table 4 indicates the Turkish replacement of the statement according to the connection type.

- The part of the sentence that is replaced for "**Application** sisteminde/n **operations on system**" changes according to the number of Entity and Application objects. This part of the sentence is composed as the following for different cases:
    - If there is only one Application "A", one Entity "E" and one connection type "C" (connection type name replaced with the Turkish statement indicated in Table 4):
      "**A** sisteminde **E** kaydı **C**(dir/dır)."
    - If there is only one Application "A", more than one Entity "E1, E2, .. En" and all connection types for this entity is of type "C":
      "A sistemi üzerinde **E1, E2, …**ve **En** kayıtları **C**(dir/dır).".
    - In the following cases where there is more than one connection type is used in an application, the statements for each connection type is formed in the order indicated in Table 4. In the partial sentence for one application, if there is at least one change or create connection type, the word "sisteminde" is added after the application name. In all other cases, the word "sisteminden" is added.
        - If there is only one Application "A", more than one Entity "E1, E2, .. En" and connection types for each entity in order is "C1,C2, … ,Cn":
          "**A** sisteminde/n **E1** kaydı **C1**, **E2** kaydı **C2**, …, **En** kaydı **Cn(**dir/dır).".
        - If the previous statement is in the form that there is the same connection type for more than one Entity which is connected to the same Application:
          "**A** sisteminde/n **E11, E21, …**ve **En1** kayıtları **C1**, **E12, E22, …**ve **En2** kayıtları **C2**, …, **E1n, E2n, …** ve **Enn** kayıtları **Cn**(dir/dır).".
        - If different connection types are applied on the same entity, the statement parts for that entity is united as the following:
          Rather than: "**E1** kaydı **C1** ve **E1** kaydı **C2**", update as "**E1** kaydı **C1** ve **C2**".
        - If the connection type "use" is used on an application, there should be at least one other connection on that application. The related sentence parts are joined using the expression "kullanılarak" and no semicolon is used between the parts in the following form:
          "**A** sisteminde/n **E1** kaydı kullanılarak **E2** kaydı oluşturulacak/görüntülenecek/silinecek …"
        - If there is more than one Application "A" and "B", and Entity objects are connected (from 1 to n) to each of them, two sentence parts formed separately for the applications are joined in the following form:
          "**A** sisteminde/n **E11, E21, …**ve **En1** kayıtları **C1**, **E12, E22, …** ve **En2** kayıtları **C2**, …, **E1n, E2n, …**ve **Enn** kayıtları **Cn**, **B** sisteminde/n **…** (dir/dır).".

- If the connections on Function-Entity pairs are tagged with numbers as selective, the statement will be formed with "or" word as exemplified as below:
  "Eğitim hizmetleri için satın almanın başlatılması sırasında, Sekreter tarafından Eğitim sistemi üzerinde Eğitim Planı veya Seminer planı veya eğitim ve çalışma takvimi kayıtları görüntülenecektir."

- Each of the sub-sentences formed by combinations of entity, application and "operations on system" part is combined as one sentence by means of comma and "and" conjunction as exemplified below:
  "Activity1 sırasında Application1 üzerinde Entity1 oluşturulacak, Application2 üzerinde Entity2 ve Entity3 değiştirilecek, Entity4 okunacaktır."

Table 4 Turkish statements for to the connection names between entity and function

Entity-Application Connection Type	Turkish Statement	Statement after application name	Verb Suffix
Uses	kullanılarak	sisteminden	-
Views	görüntülenecek	sisteminden	-tir
Creates	oluşturulacak	sisteminde	-tır
Changes	değiştirilecek	sisteminde	-tir
Reads	okunacak	sisteminden	-tır
Deletes	silinecek	sisteminden	-tir
Lists	listelenecek	sisteminden	-tir

Type-3 Requirements Sentence:

The type-3 requirements sentence structure in natural language will be composed in the following way using connection types between organizational elements and functions. This type of sentence is utilized to convert the constraints defined on FADs to requirement sentences. A separate requirements sentence is generated for each constraint on FAD. The sentences as the name of the constraint object is formed to be compatible with the requirements sentence structure. The sentence shall be finalized with one of the "-ecektir, -acaktır" phrases.

"**Activity**(me/ma) işlemi sırasında **Constraint**."

Generation of Requirements Document

The requirements sentences identified according to the procedures explained above need to be organized into a document to form the user requirements document. This section describes the structure of the requirements document.

One requirements document generated covers one modeling project. After the initial cover page, an index page is placed which include modeling project name, process map name and the list of VC, FTD, EPC and OC diagrams. In the following pages, requirements sentences are listed under the headings identified for each EPC. The name of the related process and process address is written as the heading for the process model. The process models are listed in order (in order of hierarchy, going down to lower level). Sub-headings are formed as required by the process hierarchy. Each requirements sentence is tagged with a hierarchical number under the heading, and also with a unique number increasing throughout the document (REQ1, REQ2 …). The illustration of the requirements document structured explained above can be seen in Figure 23.

n1 Process name: **“process name 1”**
Process address: **“process path 1”**
    n1. k1. REQ1: **“Type-1 requirements sentence”**
    n1.k2. REQ2: **“Type-2 requirements sentence”**
    n1. k3. REQ3: **“Type-3 requirements sentence”**
    n1. k3. REQ4: **“Type-3 requirements sentence”**
    ….
    n1.kX. REQX: **“Type-3 requirements sentence”**
n1.1. Process name: **“process name 1.2”**
Process address: **“process path 1.2”**
    n1.1. k1. REQ(X+1): **“Type-1 requirements sentence”**
    n1.1. k2. REQ(X+2): **“Type-2 requirements sentence”**
    n1.1. k3. REQ(X+3): **“Type-3 requirements sentence”**
    …

Figure 23 Organization of the requirements document

Discussion on the Generated Requirements:

Sommerville et.al. provides guidelines to describe the requirements [191]. By generating user requirements in UPROM, we claim to automatically conform to these guidelines.

- “*Define standard templates to define requirements*”
  By making the requirements sentences written in a standard way, it is easier to read, collect and write them. The sentences should follow certain structures, such as stating inputs, processing and outputs for functional requirements.
- “*Use language simply, consistently and concisely*”
  The requirements should use a simple language and express only whatever required to be expressed. More than one requirement shouldn’t be expressed in a single sentence. Acronyms and terminology should be used consistently.

In UPROM, each function is analyzed to identify how the function is expected to execute by the user in a process automation software. Thus, the generated requirements are mostly functional requirements. However, during the identification of constraints, the users can also add information on expected non-functional properties of the system. But we don't expect to identify a full set of non-functional requirements by applying UPROM. Thus, analysts should further analyze and determine non-functional requirements.

In UPROM, functional requirements regarding to specific processes and functions of the system are identified. Additional functional requirements shall also be identified which are not applicable to a specific process and function of the system, but rather focused on general functionalities of the system.

Functional user requirements include the following knowledge for the software [158]:

- Data transfer
- Data transformation
- Data storage
- Data retrieval

As seen, a functional user requirement focuses on "function", "data" and "data movement" aspects of a system. The knowledge captured in the business process models covers the function and associated data in the processes; which is the basic knowledge to identify functional user requirements of the system. What is not included in the models is the data movements applied by the functions (other than the basic input-output operations). To gather this knowledge, further analysis is conducted in UPROM by means of operations named on connections between functions and entities on FADs. This also ensures that the association between business process models and data movements are kept.

3.3.2. Generation of Functional Size Estimation

COSMIC method manual describes the functional size measurement activities in three phases: measurement strategy, mapping and measurement [125]. In each phase, specific characteristics and results regarding the system are identified. In conformance with COSMIC method, UPROM identifies each aspect of the measurement conforming to the phases of COSMIC method. Though the user requirements expressed in UPROM models are not yet as mature as software requirements, model constructs and operations conducted on conceptual system entities logically correspond to FSE concepts. The conduct of the estimation is described below organized under each phase.

Measurement Strategy Phase:

The purpose of using the UPROM FSE method is to measure the size of early user requirements for the process automation software. The scope of estimation is identified by the process models. The level of granularity is higher than the standard functional user requirement (FUR) level defined by COSMIC. The software to be developed is represented by application objects in process models. If multiple applications are placed on the models, all of them are in the scope. If a system is decomposed and components are placed on the models as applications, each component is measured separately and results are summed up.

Functional users are the organizational elements connected to the functions. External applications modeled on FADs from which data is used and to which data is sent are also functional users. Thus, system boundary lies between these functional users and applications to be measured.

Mapping Phase:

Functional Processes*:* A functional process is a "*unique, cohesive and independently executable set of data movements*" [125]. Thus, as UPROM procedure, lowest level functions in EPC diagrams which are assigned an FAD as sub-diagram are functional processes. Due to the higher level of granularity, the coverage of a functional process may be broader compared to COSMIC method. EPC notation requires the event that activates the function be modeled before the related function. Thus, triggering events are also specified in EPC diagram. Considering the COSMIC rule that a functional process must have at least two data movements (DM); an FAD must also contain at least an entity and application in addition to function, and certain type of operations must be applied on that entity as will be described below.

Object of Interests (OOI) and Data Groups (DG): Operations on entities are modeled on FAD, each entity depicting an object that is to be processed by the software. Thus, each entity is an OOI on which DMs are applied. Entities are mostly persistent, showing objects existing in the system after the function is completed. Transient objects are also modeled as entities, to depict outputs that are created for that function. DGs are not explicitly specified in UPROM, but usually realized by the explanation provided by constraint objects.

As a result of Entity Relationship Analysis [158, p. 16] conducted in COSMIC method, entity types identified by ER analysis are OOIs. But this is usually persistent data. The aim of UPROM data analysis is also identifying persistent data. According to definition of DM by COSMIC, these persistent objects are involved in Read and Write DMs. In parallel, entry and exit DMs are usually conducted on transient data, such as the result of a query on data collected from more than one entity. As the resulting data is not to be stored but just shown as a report, it is a transient data. In business processes, such transient data that appear as input and output of inquiries are depicted as inputs and outputs to the functions. On FADs, such entry and exit movements are modeled as creation of a "report", which includes entry and write events. Though many different inquiries may result in the creation of the same report in the process model (which normally would be identified as different OOIs in FSM analysis), the total number of DMs will add up correctly.

Identification of DMs will be explained in the following measurement phase.

The Measurement Phase:

COSMIC measurement method calculates the size of a system by adding up all DMs [125]. In COSMIC definition, "*DM is a base functional component which moves a single data group type*". There are four DM types: Entry (E), Exit (X), Read (R) and Write (W). Description of each DM provided by COSMIC method [125] is listed below:

- Entry (E): moves a DG from a functional user into the functional process
- Exit (X): moves a DG from a functional process to the functional user
- Read (R): moves a DG from persistent storage into the functional process
- Write (W): moves a DG lying inside a functional process to persistent storage

In UPROM, as the requirements analysis is conducted at a high level, individual DMs are not yet identified in FADs. However, operations conducted on each entity on FADs provide information about DMs of the related functional process. Using operations to determine the lifecycle and DMs of OOIs in a functional process is also suggested by COSMIC business application guideline [158]. In UPROM, seven operation types based on CRUDL operations are determined to express user requirements. Operation types used in UPROM, base CRUDL operations for them and DM conversions for each operation are provided in Table 5 below.

Table 5 UPROM conversion from operation type to DM

Operation Type	Base CRUDL Operation	Data Movements
create	create	E, W
change	update, list	E, R, W, X
delete	delete	E, W
view	list	E, R, X
list	list	E, R, X
read	read	R
use	read	R

The interpretation and rationale for using each operation type is described in the notation part of the methodology in section 3.1.5. Here, we rephrase the description of each operation type and describe its interpretation from COSMIC perspective.

- **Create:** It is used when a new instance of an entity is to be created in the persistent storage. It is also used for an entity which is connected to an external application, thus the connection with the create operation indicates that related OOI of the entity is sent to the external application.
- **Change:** It is used when the data of an existing entity need to be updated. In COSMIC method, change is usually considered as two separate functional processes: "Retrieve for update" and "update". As UPROM handles them as only one operation, 1 E and 1 X DM less is measured compared to COSMC. COSMIC mentions that these two can also be taken as a single functional process. Assuming that functional processes are analyzed at a higher granularity in

UPROM, these two are considered single operation, and single E and X DMs are added.
When there is a change operation in an FAD, it is common that the entity to be changed also needs to be listed or queried. So, the existence of list, search or query shall also be considered when change operation is used.

- **Delete:** It is used to indicate the deletion of an instance of the mentioned entity.
- **View:** It is utilized when the data of the specified entry are to be displayed for the user. Usually, additional operations are also conducted on the entity when the data are provided to the user with the view operation.
- **List:** This operation is used for listing of all entity instances (without any filtering) or listing by a query for one or more of the entity's attributes. List is also used when an automatic listing (like drop-down list population) is conducted as a result of operations on other entities.
- **Read:** It is used when it is necessary to obtain the data on the given entity. It is used when a single entity needs to be retrieved and used in other operations. The entity of which its data are read is not shown to the user. It is a supporting operation type, there needs to be operations of other types on FADs; as merely a read operation does not satisfy the conditions of COSMIC to be a functional process.
- **Use:** This operation type is also not used on its own, but must be accompanied by other operation types of view or list. E.g. when an entity is queried, use operation can be used to obtain data to filter the query.

DMs are calculated for each entity, then added up for each application based on entity-application connection on FAD. Further rules are utilized to adjust DM values calculated by converting operations based on Table 5. Before introducing the rules, an example conversion is provided below for an FAD shown in Figure 24 below.

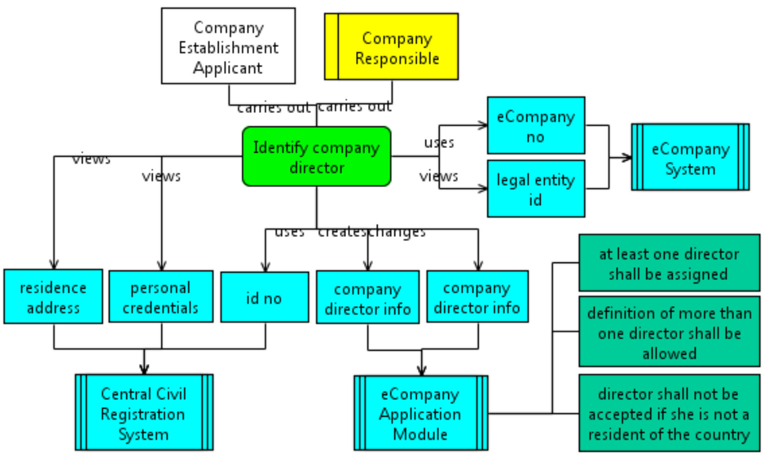


Figure 24 Example FAD for UPROM FSE illustration

- eCompany System: 1 Uses + 1 Views: 1E + 2R + 1X
- eCompany Application Module: 1 Create + 1 Change: 2E + 1R + 2W + 1X: 7 DMs

- Central Civil Registration System: 1 Uses + 2 Views: 2E + 3R + 2X: 7 DMs

The rules applied to fine-tune DMs determined by the conversion and identify total estimated functional size for each application are described below. The DMs in an FAD are calculated separately for each application on the diagram. Thus, the following rules are applied separately for each application.

FSE Rule 1: If there is a create or delete operation in a functional process but no X DM is introduced, one X DM is added. The rule is applied taking of list and view operations. The rationale for this rule is that, there shall be one X for all create and delete operations in a functional process, but not an X for each operation to prevent the improper accumulation of X DMs. Apart from that, each list and view operation shall have its own X DMs caused by basic COSMIC rules. Each change operation inherently contains retrieve and update, thus having an X DM in the conversion table. This rule also ensures that there is at least one X DM in a functional process. The rule is illustrated in Figure 25. Before applying rule 1, total DM of the functional process would be: 1 Create + 1 Delete: 2 E + 2W = 4. With Rule 1, 1 X DM is added and resulting DM is 5.

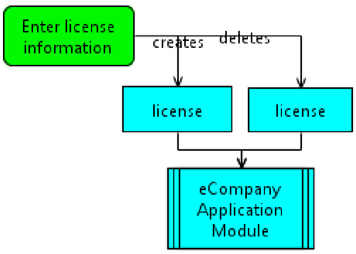


Figure 25 Illustration of FSE rule 1

FSE Rule 2*:* To prevent unnecessary accumulation of E DMs in functional process, the sub-rules described below are applied. At COSMIC method, when an OOI is listed, viewed, created, changed or deleted, DMs of type R, W and X may be applied for other related OOIs depending on the ER model of the system. If this is the situation, usually E DM is only one in the functional process, as there is only one triggering entry. But considering UPROM conversion table (Table 5), multiple E DMs will be added up for each such operation. To prevent such occurrences, ER diagram of UPROM is used to identify "related entities" and remove extra E DMs for those entities.

To illustrate how this rule is applied, consider a part of the ERD for a simple celebrity system in Figure 26. Figure 27 expresses an FAD using this ERD, where a paparazzi enters a record of a catch he discovers for a celebrity in a certain place. When direct conversion is applied for this FAD, functional process should have 3 E DMs caused by one create and two list operations. However, celebrity and place entities are listed only because the user requests to enter a catch record, thus no other E DMs are added with respect to COSMIC rules. To comply with this, UPROM method removes extra E DMs when they are introduced in the same FAD because of the operations on related entities.

E DMs are removed for operations on the entities that are assigned to be "related" by the following rules.

- Entities connected with generalization, relationship or aggregation are related.
- Entities need to be directly connected and they should be on opposite sides of the connection (decided by the direction of connections on diagrams). Only one relation needs to exist in between. Multiple levels of relations are not accepted.
- Entity is not accepted to be related to itself. Thus, separate E DMs are added for multiple operations on the same entity.

We are aware that not all of the operations on related entities are applied due to the triggering effect of the main entity. Some of the operations on the related entity may be applied because of any other independent reason, thus the application of the rule may result in inappropriate removal of E DMs. Considering that each activity and the related FAD is focused on some specific functionality of the system, we assume that such cases will occur rarely enough and will not introduce a big diversity in size estimation results.

We also assume that operations on all sub entities and their aggregate entity relevant for the functional process are explicitly modeled on FAD. If the modeler analyses the processes by only placing the aggregate entity and assumes that the same operations will also be valid on sub entities, the size needs to be automatically multiplied for all sub entities. If this is the practice, this rule must be updated in UPROM appropriately to calculate size in this way.

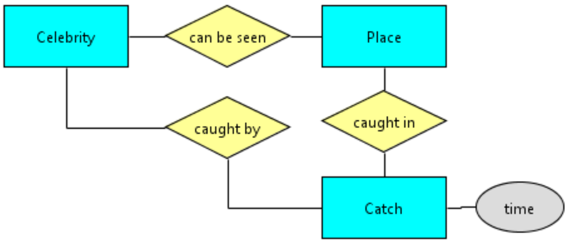


Figure 26 A partial ERD for illustration of FSE rule 2

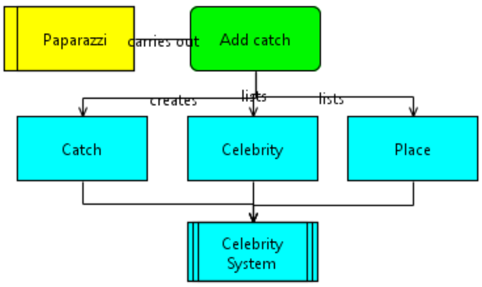


Figure 27 An example FAD for illustration of FSE rule 2

FSE Rule 3: If a function on an FAD is executed by more than one organizational element, the measured size of the functional process shall be multiplied by the number, as there are two separate functional users that can conduct the function. However, if FAD is referenced more than once in the models, the size is not added up again.

FSE Rule 4: According to COSMIC method, a functional process shall comprise at least two DMs, an E plus either an X or a W. Thus, at least one operation of type create, change, delete, view and list shall exist on an FAD. Read and use operations cannot exist alone.

The FSE rules 1 to 4 apply to operations conducted on entities residing on a single application in FAD. The Following rules are applied for the whole modeling project.

FSE Rule 5: The whole modeling project is scanned for applications for which only the following operations are conducted on its entities: list, view, read, use. It is concluded that this is an external application from which data is only requested and viewed. There may be also other external applications shown on FADs to which data can be provided, like external web services. Only create operations are conducted on those applications to indicate that data for an entity are sent. Both types of external applications are functional users of the system. DMs calculated for such applications are added up to the size of the main application in the related FAD.

FSE Rule 6: Any entity in the modeling project must be created in at least one functional process during its lifecycle. This means that at any place through all business processes, we should be observing a create operation for each entity. This is not applicable to entities residing on external applications identified by the previous rule.

To exemplify the detailed application and depict how rules are applied, an FAD is provided in Figure 28. Constraint objects are removed as they are not of interest for FSE. To implement the rules, a part of ER diagram related to the FAD is given in Figure 29.

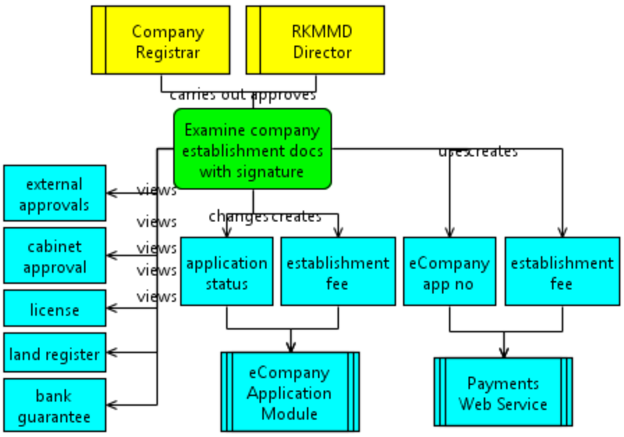


Figure 28 Example FAD to illustrate application of the full FSE rule set

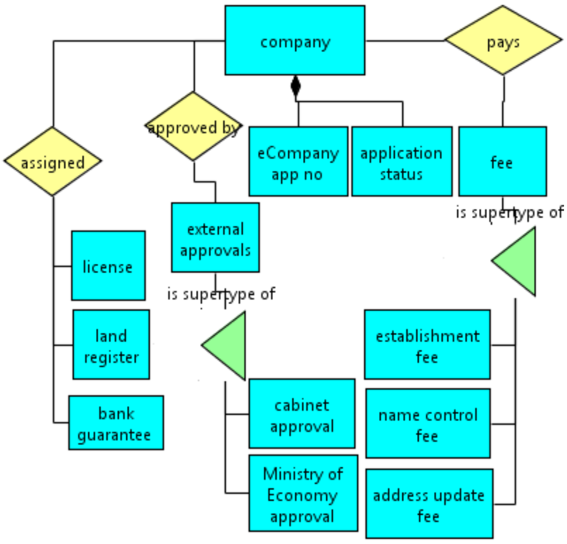


Figure 29 Example partial ERD to illustrate application of the full FSE rule set

The application of UPROM FSE method phases for eCompany system and the calculation of estimated size for FAD in Figure 28 is explained below in steps.

- The initial phase is the measurement strategy phase to be identified for the whole software to be developed. The scope of measurement is the process automation software defined by the modeling project in which FAD in Figure 28 resides, which is eCompany. eCompany system is decomposed into software modules, which are all in the scope. eCompany Application Module depicted in the figure is one of them. Functional users in Figure 28 are Company Registrar and RKMMD director. Payments Web Service, which is an external application, is also a functional user.
- Mapping is the second phase. For the FAD in Figure 28, the functional process is: Examine company establishment docs with signature. OOIs are the entities of external approvals, cabinet approval, license, land register, bank guarantee, application status, establishment fee and eCompany app no.
- The third phase is the measurement phase. The size is calculated for applications in each FAD and added up to find the overall size. The calculations are exemplified below for FAD in Figure 28.
- For each application, convert operations to DMs according to Table 5:
  Payments Web Service: 1 creates + 1 uses = 1E + 1R + 1W
  eCompany Application Module: 5 Views + 1 Create + 1 Change = 7E+6R+5W+6X
- ***Apply FSE Rule 1***: The rule is applicable only to "Payments Web Service" DMs, as no X DM exists for it although there is a create operation. Thus, total DMs for this application is updated as:
  Payments Web Service: 1E + 1R + 1W + 1X
- ***Apply FSE Rule 2***: This rule is not applicable to Payments Web Service, as the two entities "eCompany app no" and "establishment fee" are not related. For eCompany Application Module, we observe ERD considering the rules and

conclude that only two entities “external approvals” and “cabinet approval” are related. One E DM is removed for this and the resulting DMs for this application are updated as:
eCompany Application Module: 6E + 6R + 5W + 6X

- ***Apply FSE Rule 3:*** The function is executed by more than one organizational element (one with “carries out” and the other with “approves” responsibility). Thus, we multiply all DMs by two:
eCompany Application Module: 12E + 12R + 10W + 12X
Payments Web Service: 2E + 2R + 2W + 2X
- ***Apply FSE Rule 4***: FAD conforms to the constraint as there are view, change and create operations. No action is required.
- ***Apply FSE Rule 5:*** Considering the whole modeling project, we understand that Payments Web Service is an external user of the system. The main application of this FAD is eCompany Application Module. If there were more than one other application, the main application would be the one connected to the function on EPC level. Thus, we add up all DMs to only one application and end up with the following result:
eCompany Application Module: 14E + 14R + 12W + 14X
- ***Apply FSE Rule 6***: For eCompany Application Module, the entities which are viewed and changed in this FAD must be created in other diagrams. Thus, we need to check that create operation is conducted for “external approvals, cabinet approval, license, land register, bank guarantee, application status” entities in any other FADs of the modeling project.

Aggregation of Results:

DMs for the applications placed in the modeling project are found by processing all FADs as explained above. According to COSMIC method, the size in COSMIC FP for an application is calculated by adding up all DMs related to that application. In the same manner, UPROM adds up all DMs for each application calculated in individual FADs.

Generation of Functional Size Estimation Report

The data movements are identified for each application in the modeling project according to the procedures explained above. This section describes the structure of the functional size estimation report. The generated size estimation report covers one modeling project. After the initial cover page, an index page is placed which include modeling project name, process map name and the list of FTD and EPC diagrams. In the following pages, number of different DM types are depicted for each FAD under the headings identified for each EPC. At the end of this the, total estimated size in COSMIC FP for each application is given. The structure of the document is shown in Figure 30.

n1. Process name: **"process name 1"**
Process address: **"process path 1"**

n1.m1. Process name: **"sub process name 1"**
Process address: **"sub process path 1"**

- n1.m1.k1. MSR1. "FAD name"
  "Application X"
  Entry Function Point: "Entry FPs measured in this FAD for Application X"
  Read Function Point: "Read FPs measured in this FAD for Application X"
  Write Function Point: "Write FPs measured in this FAD for Application X"
  Exit Function Point: "Exit FPs measured in this FAD for Application X"
  Total: "Function Point measured in this FAD for Application X"
  "Application Y"
  …
- n1.m1.k2. MSR2. "FAD name"
  …
- n1.m1.kX. MSRX. "FAD name"
  ….

Functional Size Estimation Summary:
Application X: "Total FPs measured for this application"
Application Y: "Total FPs measured for this application"
…
TOTAL: ""Total FPs"

Figure 30 Organization of the FSE Report

Discussion on the Generated Functional Size:

"*All Functional Size Measurement methods aim to measure a size of the functional user requirements (FUR) of software*" [192, p. 10]. Considering this, as we claim that business process models are a good source to identify functional user requirements of the process automation software, the same knowledge can also be used to estimate the size of this software. COSMIC Method specifies that "data movements either [125, p. 14]:

- *Enter a data group from a functional user to the software, or exit a "data group" from the software to the functional user, or*
- *Move a data group to persistent storage or retrieve it from persistent storage*"

There are four data movement types that can exist in a software system: Entry, Exit, Read, Write. These DMs correspond to the "smallest" possible movement of data, which cannot be further broken down into movements. Usually, when the requirements engineers state the DMs conducted by functions, they do not use these atomic DMs, but use more common higher level movements which are mainly a collection of atomic movements. We identify these higher level movements as "operations" on FADs.

COSMIC Method uses "software requirements" to measure the software system and identify the size in FPs, although it uses the term "functional user requirements (FUR)".

In our methodology, we identify “user requirements”, which are high level compared to the software requirements. The data analysis to be conducted is higher level, as this is the first level of analyzing the processes and the system to be automated. We do not aim to delve into all details of the entities and define a full, normalized Entity-Relationship model. We do not consider detailed rules and guidelines provided by COSMIC Manual and basically aim to identify operations conducted by each function (functional process in COSMIC terms) to be automated. Moreover, functions in processes are not supposed to be “atomic” in terms of software. This means that during detailed software requirements analysis, each function could be decomposed to a number of functional processes for the software system. This is the reason that “functional size estimation” is conducted in UPROM rather than “functional size measurement”. COSMIC Method Manual accepts this situation and provides guidelines on “early sizing”, which are considered in this methodology. Still, it should be kept in mind that this estimation should be enhanced and developed into measurement in the ongoing phases of SDLC. In this study, we use the term “user requirements” for the functional requirements defined in UPROM in higher granularity and in an early phase of SDLC.

3.3.3. Generation of Process KPI List

Identification of process metrics, or alternatively process KPIs are critical to establish a measurable performance management system based on business processes. Business process performance is measured based on process performance indicators. We see that process KPIs can be defined in process models as there is KPI object natively in EPC notation [173]. However, definition of KPIs in the models are usually not sufficient to implement them properly on a process automation and performance management system. In UPROM to identify process KPIs accurately and provide necessary information to implement them in process automation, additional connections in the meta-model and attributes are defined as described in section 3.2.1.

process#. Model Name						
process address						
KPI	Description	Related function	Related event	Information sources	Collection period & time	Hypothesis
Name of the KPI object. All KPIs in the process are listed.	*“description” attribute of KPI object.*	*The function KPI is connected to (if exists). KPI is to be measured during the execution of that function.*	*The event KPI is connected to (if exists). KPI is to be measured on occurrence of that event.*	*The information carrier(s) KPI is connected to (if exists). KPI is to be identified by using those sources of information.*	*“collection period and time” attribute of KPI object. KPI is to be measured accordingly if not identified by function and events.*	*“hypothesis” attribute of KPI object. It shall be used to analyze collected KPI values on process execution.*

Figure 31 Process KPI list template

Information on KPIs that can also be observed from the models is obtained as a report in UPROM. This list can be utilized to observe all KPIs related to the modeling project and as requirements on KPIs to be collected by the process automation software. The template used to generate the process KPI list report is provided in Figure 31. KPIs are listed on a single document under tables organized for EPCs. The fields to be completed by the information from the models is entered in small and italic font.

3.3.4. Generation of Process Definition Document

The process definition document of UPROM presents natural language definitions of processes in a predefined template format. A template is prepared by examining content descriptions from standards [24]–[26], [159] and example process definition documents from our experiences in many organizations [193]. A separate numbered section is composed for each VC, FTD and EPC diagram in the modeling project. Those sections are concatenated to form the whole document for the modeling project.

The templates used to generate the document is provided in the figures below. The fields to be completed by the information from the modeling project is entered in small and italic font. Figure 32 provides the template used for VCD and FTD. Figure 33 provides the template used for EPC.

process#. Model Name

process address

1. Process Information

Process Purpose: *purpose field in process meta-data*

Process Scope: *scope field in process meta-data*

Process Status: *status field in process meta-data*

Process Version: *version field in process meta-data*

Process Author: *author field in process meta-data*

Description field in process meta-data placed as a paragraph.

2 Activities

2.1 *Name of VC/Function Object on the Diagram*:

Sub-process: *sub-diagram attribute of the VC/Function object (if exists)*

Products: *product(s) connected to VC (if exists)*

Risks: *risk(s) connected to VC (if exists)*

Description attribute of the VC/Function object placed as a paragraph.

…

2.X *Name of VC/Function Object on the Diagram*:

…

Figure 32 Process definition document template for VCD and FTD diagram types

process#. Model Name

process address

1. Process Information

Process Purpose: *purpose field in process meta-data*

Process Scope: *scope field in process meta-data*

Process Status: *status field in process meta-data*

Process Version: *version field in process meta-data*

Process Author: *author field in process meta-data*

Description field in process meta-data placed as a paragraph.

2 Process Responsibilities

		Responsibility Type				
Responsible	Type	R	A	S	C	I
Name of the organizational element object. All organizational elements connected to functions in the process are listed.	*Type of organizational element (organizational unit, group, position, external person, location)*					
		Marked for the connection name between organizational element - function				

3 Inputs

Name	Type	Source	Link
Name of the information carrier object. All information carriers which are inputs in the process are listed.	*Type of information carrier (document, list, log, file, product, reference)*	*Placed if input is provided by another role. Name of the organizational element connected to information carrier as input*	*Document link attribute of the information carrier*

4 Outputs

Name	Type	Target	Link
Name of the information carrier object. All information carriers which are outputs in the process are listed.	*Type of information carrier (document, list, log, file, product)*	*Placed if input is sent to another role. Name of the organizational element connected to information carrier as output.*	*Document link attribute of the information carrier*

5 Entrance Criteria

Event	Processes that exit with this event	Address
Name of the event object. All events without an incoming control flow connection are listed.	*Name of the other EPC diagram(s) in the modeling project which has this event as its finish event. Shortly, other processes that exit with this event.*	*If exists, the address of the process(es) that exit with this event*

5 Exit Criteria

Event	Processes that start with this event	Address
Name of the event object. All events without an incoming control flow connection are listed.	*Name of the other EPC diagram(s) in the modeling project which has this event as its start event. Shortly, other processes that start with this event.*	*If exists, the address of the process(es) that start with this event*

(continued in the following page)

7 Activities

All activities in the related EPC diagram are presented here in a numbered list in the order of vertical placement.

7.1 *Name of Function/Process Interface Object on the Diagram*:

Responsible(s): *if it is a process interface:*

- (external process)

Referenced process: Address of the sub-diagram attribute for the process interface (if exists).

If it is a function with an EPC/FTD assigned as sub-diagram:

- (sub process)

Sub process: Address of the sub-diagram attribute for the function.

Otherwise:

For all organizational elements connected to the function:

Name of the organizational element object – name of the connection (carries out, approves, supports, contributes to, must be informed on completion).

Inputs: *The list of input information carrier objects. Placed if any input exists.*

Outputs: *The list of output information carrier objects. Placed if any output exists.*

Application: *Application object connected to the function (if exists).*

Risks: *risk(s) connected to the function (if exists).*

Description attribute of the function/process interface object placed as a paragraph.

Business Rules: *business rule(s) connected to the function (if exists). Provided as a list.*

…

7.X *Name of Function/Process Interface Object on the Diagram*:

…

8 Business Rules

Business rules in the related EPC diagram are presented here as a list.

- Name of the business rule

Activity: Name of the activity the business rule is connected to.

9 Other Processes Referencing This Process

All processes that reference this process as a process interface are listed here together with the process address. In this way, the processes using this process can be observed.

10 External/Sub Processes Used by This Process

External processes referenced with a process interface and sub-processes assigned in sub-diagram attributes are listed in two separate headings here.

External Processes:

- List of external processes referenced with their address.

Sub Processes:

- List of sub processes referenced with their address.

11 Process KPIs

The table structure in the generated "Process KPI List" is used.

Figure 33 Process definition document template for EPC diagram type

As seen in the templates, process definition document provides a standard list of many process elements in natural language format. The information is structured into lists and tables as much as possible to enhance readability. Many of the headings are standard fields used in the documents such as responsibilities, inputs, outputs, entrance and exit criteria and activities. Such information is enriched with details such as source and link for the inputs; target and link for the outputs; detailed activity descriptions including inputs, outputs, responsibilities, sub-diagrams, risks. Such detailed information is either not achieved in traditional documents, or it is hard to provide and maintain the detailed information in a standardized way for every process element [160].

Some fields in the process definition document provide cross-process information which cannot be observed by examining the single process, but can be obtained by analyzing the relations across the processes. This is especially valuable for this document, as in this way additional information is provided on the processes which cannot be observed from the individual process models. Such fields are;

- For the entrance criteria, the processes that exit with this event
- For the exit criteria, the processes that start with this event
- Other processes referencing this process

As observed from the templates, detailed information added to the models by means of attributes are utilized heavily in the generation of the document. Such descriptions are also important to give the document a "natural" feeling rather than a mechanical structure. The document has characteristics close to a manually written one while it is assured that all the fields of the document are filled in a complete and structured way.

The section on the activities constitute the longest part of the document. Typically, process definition documents include a list of activities ordered in a rough timely manner and control flow relations between them are specified in natural language descriptions. Similarly, the template also includes a list of activities listed according to the vertical placement of activities. However, the information on alternative paths and parallel executions due to the logical operators are lost in the document. One must utilize the models to observe the details of the control flow, which is obviously a less complex representation for alternative flows compared to natural language. Still, as an alternative, the template can be enriched with suggested techniques to compose natural language texts from control flow elements can be used [164].

We observed the following differences in process documents when compared to typical manually developed documents from our experiences [193]:

- In some of the documents, there is a "Definitions and Abbreviations" heading at the beginning. We aim to cover it by providing a system based business glossary, rather than one specific to the process.
- Process models are usually embedded in the document. We did not embed because we already generate another report for the models. However, this could easily be added if requested.
- The information on the responsible of activities is usually expected to be found on the definition of the activity. However, this is one of the key information a process model must specify. On UPROM's generated process definition

document, it is assured that responsibility information is provided and shown on a separate line.

- We don't see examples where responsibilities, application systems and business rules of one activity is stated explicitly for an activity, but we usually observe that this type of information is provided on the description of the activity. In some documents, inputs and outputs for each activity is provided in a separate table, but usually this information is also embedded in the descriptions. We think that depicting those fields explicitly help to better define the activity.
- Type of the input/output is not mentioned in the documents we examined. It may be important to identify the type, however, if most of the inputs/outputs in the system are of same type, that column may be omitted. In most documents, references are shown under a separate heading. In our document, we provide this information with the "type" column.
- We do not observe information on the "link" of the inputs/outputs used. But from our discussions, we think that this lack of observation is caused by the problems in maintainability. Most of the process implementers would like to see how they can reach the related input/output.
- In some documents, roles and responsibilities are detailed, rather than providing just the name of the role. This could be achieved in our documents by adding a technical term description to an organizational element.
- In many documents, sections for approval list and version history are added. Generated document lacks this part, as this is rather specific to the company. These pages should be added if required.
- Only few documents define the points in the system where the related process is used, and the list of process interfaces and sub-processes referenced by this process. Also, many of the documents provide information on "entrance and exit criteria", but none on the preceding and following processes caused by these criteria. We observe that this type of information is critical to see the placement of the specific process as on a process map. They are usually avoided in manually developed process documents, as it is very hard to maintain every time any other related process is updated. By means of automated generation, the difficulty is overcome.

3.3.5. Generation of Business Glossary

Business glossary is a supportive document to ensure common understanding of the terminology used in the processes. It is an important artifact to unify perception of the processes in business and technical domains [194]. Business glossaries are frequently prepared, but become out of date as they are not updated with the changing organizational environment. By capturing the terminology within the models in UPROM, updates on the terminology are easily conducted. Moreover, as the objects are stored as unique elements, single definition is maintained for each term throughout the modeling project. Modelers insert the terminology definitions by means of "technical term" attribute for product, risk, application, organizational elements, information carriers, entity, cluster, attribute and key attribute constructs as described in section 3.2.2.

Moreover, when the definitions inserted in these constructs are not sufficient, additional terminology can be added by means of technical term construct.

The data elements compose the most important part of a glossary [195, Ch. 8]. The entities and information carriers are placed in the third part of the generated business glossary in a separate table and formatted to reflect their relations between each other. In addition, application and organizational element definitions are frequently discussed in the organizations. Thus, the business glossary template includes three different parts to provide separate lists for applications, organizational definitions and entities.

The template used to generate the business glossary is provided in Figure 34. Business glossary items are listed on a single document for the whole modeling project. The fields to be completed by the information from the models is entered in small and italic font.

1. Organizational Definitions

Term	Abbreviation	Definition
Name of the organizational element object which is assigned technical term attribute. All organizational elements are listed in this table.	*The abbreviation mentioned in technical term attribute (if exists).*	*Technical term attribute of the related object (if exists).*

2. Application Definitions

Term	Abbreviation	Definition
Name of the application object which is assigned technical term attribute. All applications are listed in this table.	*The abbreviation mentioned in technical term attribute (if exists).*	*Technical term attribute of the related object (if exists).*

3. General Definitions

Term	Abbreviation	Definition
Name of the entity/cluster in ERD. ERD is scanned for high level entities composed of parts connected with aggregation relation. First, high level entity is placed in the table. Then, parts of that entity are placed in the next row indented one level right. Indentation goes on through the lower levels. *For entities connected with relationship t, initially the single entity in one side of the relationship is placed in the table. In the next row, relationship name is placed indented one level right. In the following rows, entities in the other side of the relationship are placed each indented one level right.*	*Abbreviation mentioned in technical term attribute (if exists).*	*Technical term attribute of the related object (if exists). If the entity has sub-types (connected with a generalization object), they are listed here as:* Sub types: *name of sub-type elements.* *If attribute objects are connected to the entity, they are placed here as a bulleted list.* Attributes: *name of the attributes*
Rest of the objects with technical term definitions (information carrier, product, risk, attribute, key attribute) are scanned. If they are not already placed in the table, each is added in a new row.	*Abbreviation mentioned in technical term attribute (if exists).*	*Technical term attribute of the related object (if exists).*

Figure 34 Business Glossary Template

3.3.6. Generation of Process Improvement List

Business process improvement is one of the major purposes for BPMod. There are detailed techniques to identify differences between as-is and to-be models and specify the improvements. During BPMod in most circumstances, the modelers need a mechanism to define the improvements they identified in natural language but related to the business processes. To fulfill this need, improvement construct is used in UPROM.

Process improvement objects that can be placed on FAD and EPC provide an easy-to-use mechanism to store improvements during modeling activities and related to the models, as described in section 3.2.1. It is important to identify the improvements during modeling activities, as it is hard to specify them after the completion of analysis. Improvement object can be used in EPC and FAD. Improvement objects can either be connected to a function, or assigned as a general improvement for the whole process when not connected to a function.

The improvements identified during modeling can be obtained as a documented list organized for diagrams and functions. This document can be used to evaluate the results of modeling activities and especially to present to the management. Process improvement list is generated conforming to the template provided in Figure 35. Improvements are listed on a single document for the whole modeling project. The fields to be completed by the information from the models is entered in small and italic font.

Process Improvements List

Name	Description	Function	Diagram
Name of the improvement object. All improvement objects are listed in this table, grouped for the same functions and diagrams.	*The description attribute of the improvement object.*	*The name of the function object the object is connected to (if exists).*	*The address and name of the diagram the object is placed in. If it is the same with the previous item, it is left blank.*

Figure 35 Process Improvement List Template

3.4. UPROM Tool

UPROM tool is developed as a prototype to apply UPROM as the methodology. It is a graphical BPMod tool that supports UPROM and automatically generates the artifacts identified by the methodology. UPROM tool is developed on Eclipse Platform using Java coding language. Model driven development approach is followed in the development [196] by using Eclipse Modeling Framework (EMF) [197] and Eclipse Graphical Modeling Framework (GMF) [198]. UPROM tool is developed as plugins for each editor.

UPROM tool provides editors for the six diagram types identified by UPROM. The editors are based on the common meta-model of UPROM as described in section 3.1. The constraints for connections defined in the meta-model are also applied. UPROM tool is developed based on bflow* Toolbox that supports EPC and VC modeling [116]. Thus, special features of this tool such as continuous verification and easy modeling features are inherited. UPROM tool also works in integration with the configuration management tool, SVN. The files of the modeling project are kept under configuration and versioned.

A typical modeling environment of UPROM tool and its components can be seen in Figure 36. Specific features of UPROM tool that supports the methodology are described below.

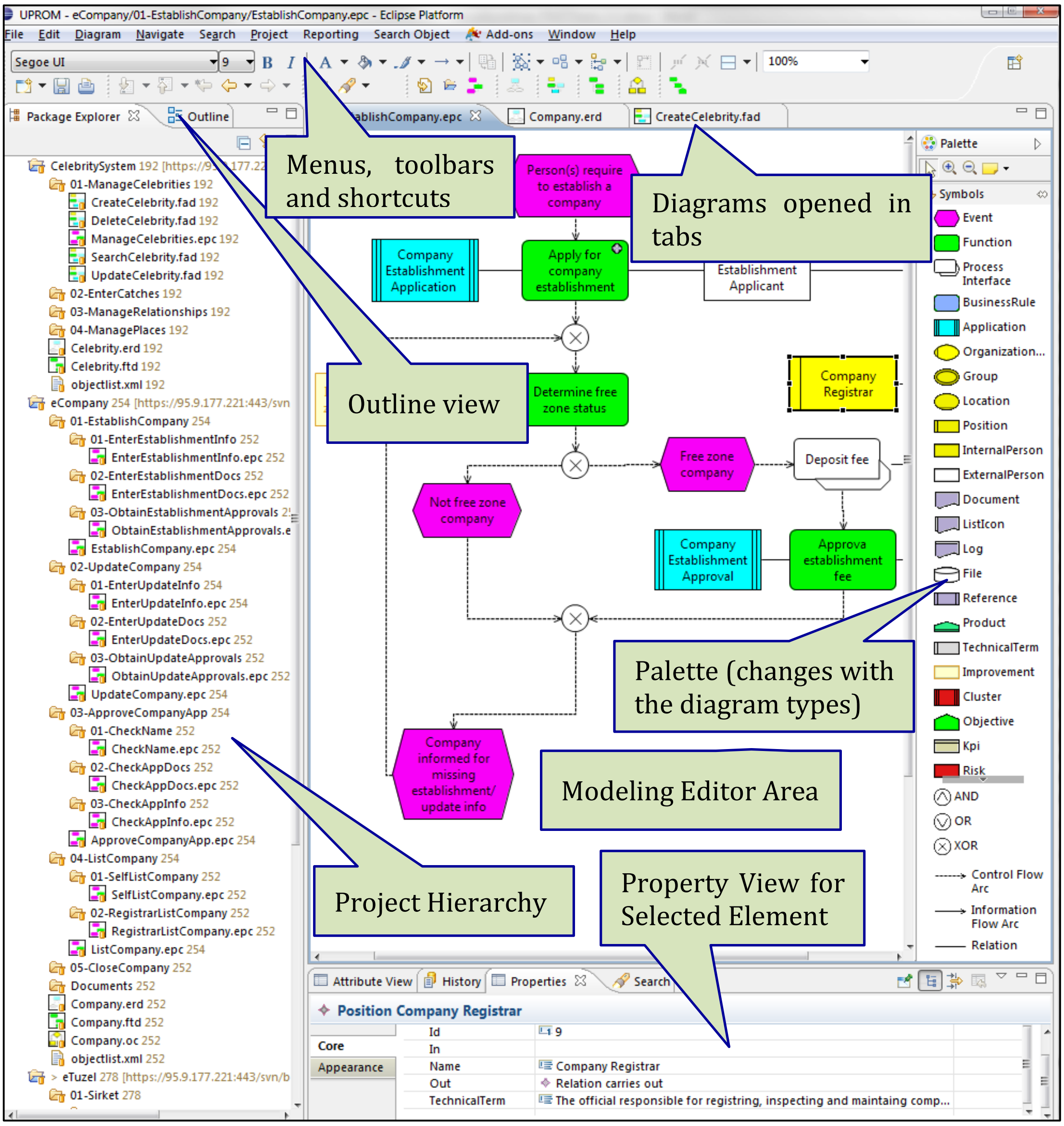


Figure 36 Typical modeling environment for UPROM tool

Connection restrictions for meta-model:

Relations between constructs are restricted in conformance with the meta-model in the modeling editor. The tool prevents formation of a connection not allowed and informs the user with a sign as shown in Figure 37.

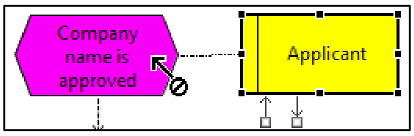


Figure 37 Disabled connections based on the meta-model

Restriction on assignment of connection names:

For the connections which are specified to have certain names in the meta-model, the tool forces the user for selecting the names from the predefined list. Two examples are shown in Figure 38. First one is the connection assigned between an organizational element and function to show the responsibility of the role, and the second is between function and entity to specify operations conducted on the entity by the function.

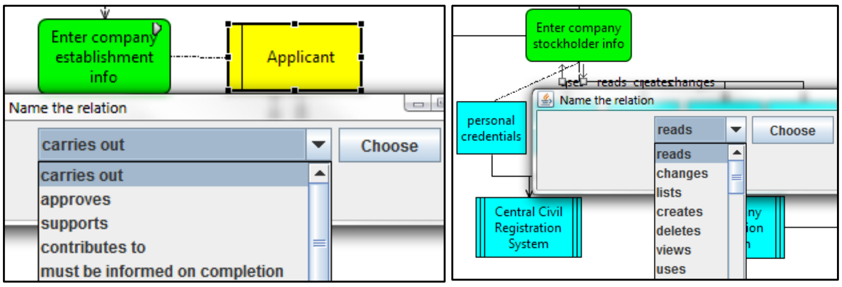


Figure 38 Assigning connection names by selecting from the combo box

Unique object assignment:

The uniqueness property is ensured by UPROM tool by maintaining the objects of the same type (or one of the alternative types) and name. Instances of an object can exist at any diagram regardless of the diagram type. When a new object is added to the modeling project, if there is an object to be unique within the project, the user is asked if the new object is the same with existing object(s) as shown on the left of Figure 39. If the user approves that they are the same, same ID's are given and their attributes are assigned to be the same. When an attribute of any instance of a unique object is updated, the updated value is reflected to all of its instances in all diagrams. Users can search for the occurrences of an object with a screen shown on the right of Figure 39, and obtain a list such as the one shown on the left.

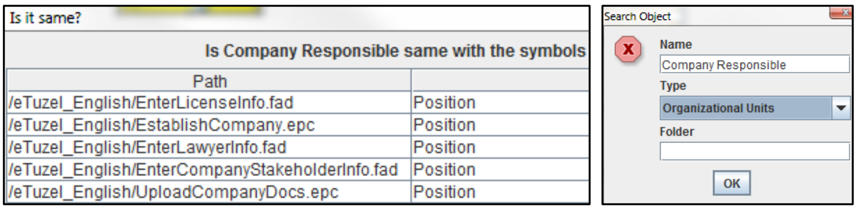


Figure 39 Assigning unique objects and search function

Process meta-data and object attributes:

Process meta-data as described in UPROM process can be assigned on UPROM tool by using the "attribute view". An example screen for process meta-data can be seen in Figure 40. The attributes for the selected object, depending on the type of the object as defined in the meta-model, are shown and assigned from the "properties" view as shown in Figure 41. For the constructs having the link attribute, if a link points to an office or PDF document, they can be automatically opened inside the tool.

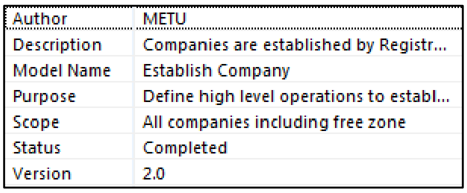


Figure 40 Process meta-data assignment

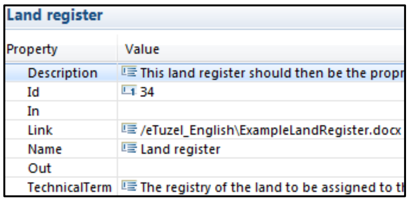


Figure 41 Assignment of object attributes

Validation for the structure of the modeling project:

The structure of the modeling project can be controlled by the tool to conform to the guidelines in section 3.2.2. The user can select the functionality from the menu, or the tool automatically checks the structure before the generation of an artifact. If the modeling structure is not validated, the tool does not generate the artifacts.

Sub-diagram assignment

Predefined sub-diagrams can be assigned tothe objects as identified in the meta-model. Sub-diagram selection and assignment functionality can be found under the right-click menu for the object. If a VCD, FTD or EPC is assigned as a sub-diagram, a small triangle is placed on the function object as shown on the left of Figure 42. If the function is allocated for automation and FAD is assigned, a small plus sign is placed as shown on the right.

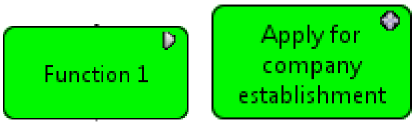


Figure 42 Illustration of functions with sub-diagram assigned

Generation of artifacts:

At any time, the users can generate the artifacts by selecting the related menu function. All artifacts are generated in PDF format. The tool parses the diagram files which are composed by the Eclipse Platform conforming to standard XML file format. The tool implements the algorithms so that the artifacts are generated as specified by the artifact generation procedures in section 3.3. It then utilizes iText library to form the pdf file conforming to the defined templates [199].

For each artifact, the tool prepares a cover page including project name, version date and author. Related information is obtained from the user by means of data entry screens when the user prompts to generate an artifact.

In addition to the artifacts generated as described in section 3.3, UPROM tool also generates PDF reports of descriptive business process models and analysis models. Business process models report includes VCD, FTD and EPC diagrams in the modeling project. The report is organized according to the process hierarchy. Name and address of the process are placed as the heading for each diagram. OC diagram is placed at the end of the report. Analysis models report provides FADs organized under the headings for EPCs and ERDs at the end.

Until now, UPROM tool is utilized successfully in all case studies of this study and in other applications. All of the referenced introduced in the following chapter are generated by using the UPROM tool.

CHAPTER 4

APPLICATION OF UPROM

This chapter presents the application of UPROM in a multiple case study setting. Section 4.1 summarizes the research approach of this study, specifies the rationale for selecting the case study research methodology, describes how the cases are selected, identifies the case study questions and propositions general to the case study research and presents a general plan. In sections 4.2 to 4.5, the conduct of the four case studies are described. Section 4.6 discusses the multiple case study results and threats to validity.

4.1. Multiple Case Study Design

The research strategy followed in this study is qualitative research. The research proposed in this study conforms many of the qualitative research properties [200]. For this study, we need to collect data in its natural setting, researcher is the key instrument in collecting the data, there are multiple forms of data and we need to conduct inductive data analysis.

At the initial research phase of this study, we conducted an extensive review of the literature on the field. This step was very important as the current studies guided this study, and the methodology developed is based on existing approaches. Moreover, due to multiple fields spanned by the dissertation topic, it requires much effort to master existing studies and associate different results with each other in these various fields. The results of the literature review are important "to fill in gaps in the literature and extend prior studies; provide a framework for importance of the study and benchmark for comparing the results" [200].

Conducting review of the literature and utilizing academic resources are activities going on through the whole research lifecycle to mature the studies in all stages. However, it is mostly critical at the initial research phase to achieve benefits stated above and establish a baseline for the Unified BPMod methodology to be developed.

Case study research is selected as the qualitative method. Case study research is the "most common qualitative method used in information systems" [200]. Case study research is appropriate in many ways to answer the research questions and propose a solution relevant to the purpose of this study.

Case study research enables to study the phenomenon "*in its natural setting, learn about the state of the art, and generate theories from practice*" [201]. It also allows "*to understand nature and complexity of processes, by answering "how" and "why" questions*" [202]. Case study "*copes with technically distinctive situation in which there will be many more variables of interest than data points*" [201] and it relies on multiple sources of evidence. Additionally in case study, "*one or few entities are examined; no experimental controls are involved*" and "*complexity of unit is studied intensively*" [202]. Likewise, focus of this study is a contemporary issue discussed within its natural domain without any manipulation. Many variables are to be examined and new ones are explored through intensive research. In this way, the issue will be tried to be explored in detail, and the appropriate methodology to be developed. "*Case studies do not generate the same results on e.g. causal relationships as controlled experiments do, but they provide deeper understanding of the phenomena under study*" [203].

Different researchers reach a similar description of case study being an investigation of "contemporary phenomena" in their own context [203], where the boundary between the phenomena and the context is not necessarily clear. This is usually the case for information systems research. Case studies are mostly used for exploratory purposes, to reveal the occurrences in a setting; but still may include explanatory perspectives to search for an explanation of a problem [203]. The usage of the case study research in software engineering usually focuses on improvement, in a similar manner to action research [203].

Our research approach includes both exploratory and explanatory characteristics. By means of the exploratory approach, the business process modeling and requirements analysis domain is studied, what is happening to conduct these activities are understood and new ideas are generated to unite these domains. Especially initial case studies in our research focused more on exploratory aspects. Another purpose of our research is being "improving" (as defined by [203]) looking for ways to improve business process modeling and requirements analysis so that we can end up in more efficient ways of conducting work in these domains.

To evaluate the applicability of our methodology in different cases, to collect more data to be able to answer our research questions and to deal with the problems of validity in case study research, we applied multiple case study research. We mostly used five of the most common six sources of evidence for the case studies to collect data: "documents, interviews, direct observation, participant-observation and physical artifacts" [201], as appropriate in different case studies we conduct. This is especially important to overcome construct validity and reliability problems. In addition to qualitative sources of evidences such as observations and interviews, quantitative source of metrics are also collected as it is important to support case study evidences with metrics [203]. Sources of evidence used at each case study is described in the related section. However, for all the case studies, we need to mention that we utilized a tool to develop physical artifacts:

UPROM tool by means of which we can automatically produce outputs as mentioned in our methodology. External validity and reliability actions are taken to ensure that this tool produces outputs as specified by the methodology.

In case 3, we conducted the case study by purely observing the evidences developed by analysts not involved in our research. In case study 2, we were involved in the application of UPROM as part of an analysis team. Activities of case study 1 and case study 4 were conducted by the researchers of this study. Thus, some case studies indeed contain the characteristics of an action research. Action research is "closely related to case study" where the researcher is involved in the activities in contrast to case study in which researcher only observes [203]. Being close relatives, the conduct of both research methodologies are similar. We report all cases in our research as part of a multiple case study research. However we are aware that our research carries also the characteristics of action research and evaluate the threats to validity considering this.

UPROM as a methodology unites many disciplines and it is very hard to find the same conditions for different cases. Still, we try to find out formal propositions that can be applicable for every case, measure different variables and answer our research questions and generalize the results for other possible cases. Thus, we follow a positivist approach in this respect.

4.1.1. Selection of the Cases

To find the appropriate cases to answer our research questions, each of the cases needs to conform to some criteria for the system under study. The following criteria were considered while selecting the cases.

- The system under study shall have complex process sets that contain relational, hierarchical and cross-referencing processes and with properties like conditionals, loops, constraints, inputs/outputs, business rules.

Such characteristics are critical to be able to propose solutions for a wide variety of conditions. This property will enhance the generalizability of the results to other cases. This property is also critical to contribute to the belief of the organization on the necessity of the studies for defining business processes. Without such a belief, the study might be perceived as a burden as part of the daily studies of the organization. Only then, does it make sense for the organization to allocate resources for the study. This consideration is significant, since the success of the work on defining the business processes is highly dependent on the contribution of the domain experts besides BPM experts.

- There exists the need to develop information systems to automate the business processes of the system under study.

The organization may be planning to develop information systems by its own resources or by subcontracting. In each way, the organization shall be aware that they need to elicit requirements to develop the system or to use as a technical contract document. They shall also think that size estimates for the systems to be developed will be helpful for them for planning the development activities or for subcontract management.

We identified two different types of cases that provide different perspectives to answer our research questions.

New implementation case: A new project is being initiated to automate a set of business processes. The organization is willing to use UPROM to model the business processes, obtain requirements, use early size estimation for project planning and utilize process definition documents to disseminate the knowledge. Thus, UPROM can be followed in this project to conduct the project activities from scratch. The results will be obtained by analyzing the artifacts generated by UPROM.

Retroactive case: A project to automate a set of business processes is already conducted. Some or all of the information that are aimed to be captured by UPROM artifacts (like process models, process definition documents, requirements, and business glossary) are already defined. The organization is willing to provide resources to re-conduct the analysis and modeling activities by using UPROM and evaluate the results. In this case, data collection and analysis are focused on comparison of activities and artifacts developed by UPROM and the native ones.

Case study 2 and 3 conform to the definition of new implementation case, and case study 1 and 4 are retroactive case type.

Each of our cases has its own characteristics, its own needs and problems to be solved and different way of producing outputs. Thus, we need to analyze, plan and conduct the study for each case differently. Our unit of analysis is the cases itself; but in the end, they all serve for the purpose of answering research questions of our study.

Considering that our methodology spans a wide variety of areas, we decided that we can also conduct case studies focusing on certain aspects of the methodology, so that we can collect bigger amount of data and reach more generalizable conclusions. Case study 1 focuses only on the functional size estimation practice, and case study 2 focuses on business process analysis and documentation practices. Cases 3 and 4 implement all aspects of UPROM.

4.1.2. Research Questions and Propositions

The research goal of this study is to investigate if business process modeling can be conducted in an integrated way for diverse practices and preparation of artifacts for those diverse practices can be enhanced. To achieve this research goal, the research questions we try to answer by means of multiple case study research and the respective propositions are as follows:

RQ1: Can we develop a methodology to analyze and model processes, generate process documentation, develop user requirements and estimate the size for the software to be automated all together and based on business process models?

P1: UPROM methodology and the related tool can be used to analyze and model the processes, develop user requirements, estimate the size of the software to be automated, generate process definition document, business glossary, process improvement list and process KPI list. All artifacts can be developed by using the information captured by business process models and automatically by the tool that supports UPROM methodology.

RQ2: Are the related artifacts better than the ones developed with traditional methods in terms of consistency, completeness, maintainability and traceability?

P2: By using UPROM methodology to define business processes:

- P2.1: business process models are developed in a well-structured way,
- P2.2: most of the functional user requirements are obtained in a consistent, complete, maintainable and traceable manner,
- P2.3: software size is estimated early and reasonably accurate,
- P2.4: process KPI list associated with process models is obtained,
- P2.5: process definition documents are developed capturing all necessary information and in an easily maintainable way,
- P2.5: business glossary is generated so that consistent usage of terminology among the processes is assured,
- P2.6: improvements obtained during analysis and modeling activities are easily captured and reported,
- P2.7: when required, updates to any process and requirements information are done on the models and all artifacts can be regenerated without any additional effort; thus maintainability of the artifacts is enhanced and they are ensured to be traceable to the single source of process models.

4.1.3. Case Study Plan, Data Collection and Analysis of Data

For each of the case studies below, the case study plan, data collection and analysis is provided in their own sections. For the multiple case study, the following general plan is followed:

- Conduct "Case Study 1 Set for Size Measurement" to evaluate the size measurement methodology suggested by UPROM. For three small business application systems, develop the models and generate UPROM size estimation results. Compare the results to COSMIC measurement values previously identified. Update and fix the size measurement methodology to be used in the rest of the case studies. This is a retroactive type case, where the results obtained by UPROM are compared by the native outputs.

- Conduct Case Study 2 Set for company processes of e-LegalEntity (Company Central Registration) system and trademark processes of e-Trademark (Trademark Central Registration) system. Develop business process models and analysis models, generate user requirements, add any additional requirement needed to form the technical contract, generate size measurement report, develop process definition document and business glossary. Collect the data and evaluate the results.
  This is a new implementation case type case study where a new project is initiated and UPROM is used to prepare deliverables of the projects. The work of this case study is conducted within an ongoing project, so the results obtained by applying UPROM are used as project deliverables and evaluated by analysts of the contracting organization. The study in this project is composed of analysis of two separate systems: company and trademark; so it is conducted as a multiple case study in itself, for which activities are conducted in parallel.
- Conduct Case Study 3 for public investment processes within the project of research and development of information map project (BİHAP) for Ministry of Development.
  The project's main focus is development of a comprehensive ontology for the ministry. A process modeling project is conducted for public investment processes of the ministry. The project's scope does not cover automation of the processes. However, within the scope of the project, business process analysis, modeling, development of process definition document, business glossary and improvement list report exist. These activities are conducted following UPROM and using UPROM tool. Because of this, this is a new implementation case study. This case study is conducted for business process analysis, modeling and generation of process document practices of UPROM.
- Conduct Case Study 4 for selected process modules of university's Integrated Information System project. This is a retroactive case study where business processes are already analyzed and modeled, process definition documents are defined, requirements are identified business processes are automated. For this case, we apply UPROM to re-analyze and model processes and generate artifacts. We compare the generated artifacts with the process definitions natively produced in the project.

Plan for each of these case studies, data collection and analysis are provided below in the sections. As we conduct case study research, we basically collect qualitative data and conduct qualitative analysis to evaluate the proposed methodology and answer the research questions. However, where applicable, it is necessary to support case studies with quantitative data collection. For this purpose, we followed a GQM approach to identify the metrics to be collected during the conduct of the case studies [203]. The research goal of the study and related research questions are provided in the previous section. The metrics to be collected for the related research questions and propositions are provided in Table 6.

Table 6 Metrics and related research questions

Question/ Proposition	Metric	Related Case	Purpose
RQ1/P1	General BPMod metrics: deepest level in the hierarchy; # EPC, FTD and FAD diagrams, # functions on EPCs, # functions per EPCs, # FADs per EPCs, # unique control flow nodes, # unique control flow nodes without event nodes, average # of total control flow nodes per EPC diagram (without events), average # of connector nodes per EPC diagram, average # of arcs per EPC For each EPC: number of functions, connectors, process interfaces, control flow nodes w/o events, events, control flow nodes, applications, organizational elements, information carriers, business rules, arcs, start events, end events.	All	To provide information on the project outputs To depict characteristics of business process models of the case. To show applicability and generalizability of UPROM for cases with different characteristics. To evaluate the conformance of the models to general process modeling guidelines in the literature.
	General Requirements metrics: total number of generated functional requirements, total number of requirements generated from constraints, % of requirements generated from constraints, number of modules, number of requirements per module, number of requirements per EPC	Case 2, 3, 4	To provide information on the project outputs To depict characteristics of requirements of the case. To show applicability and generalizability of UPROM for cases with different characteristics.
	Functional size estimation metrics: FP size for each application	Case 1, 2, 4	To provide information on the project outputs To depict characteristics of estimated size of the case. To show applicability and generalizability of UPROM for cases with different characteristics.
	Process documentation metrics: % of process models for which process metadata are added, % of functions for which descriptions are added, number of entries in business glossary, ratio of business glossary entries per total number of entities in the project, number of business glossary entries per EPC diagram, number of improvements, % of improvements per application, number of improvements per EPC, process definition document page number	Case 2, 3, 4	To provide information on the project outputs To depict characteristics of process documentation of the case. To show applicability and generalizability of UPROM for cases with different characteristics.

RQ1 and RQ2	# items in the native process definition document, # items covered by UPROM artifacts in native way, # items covered by UPROM artifacts in an adapted way, # items identified as N/A, # items not covered by UPROM, # items adapted or not covered but that can be added to UPROM	Case 4	To evaluate if UPROM can be used to generate required information as specified by existing documents To evaluate if the generated artifacts are better than the native project documents in terms of completeness
RQ2/P2.2	Metrics for manual requirements: Number of requirements added manually, % of generated requirements	Case 2	To evaluate the completeness of generated requirements
	Metrics for manual requirements: Total number of requirements in existing process definition document, number of existing requirements covered by the generated requirements document, % of requirements coverage	Case 4	
RQ2/P2.3	Distribution of data movement types	All	To evaluate if the distribution is appropriate for business application domain
	Deviation of estimated and measured functional size	Case 1 and 2	To evaluate accuracy of estimated size
RQ2	# items for which enhancements are identified by expressing the information in UPROM # of improvements	Case 4	To evaluate if the artifacts generated by UPROM are better than the native project artifacts in terms of consistency and completeness

4.2. Case Study Set 1 for Size Measurement

In this section the first case study of the study is explained. First, the background information for the three cases is explained. Case study questions are defined. Then, the plan for the case study conduct is given. Sources of evidences are identified. In the last sections, conduct of the case study and data collection and analysis results are provided.

4.2.1. Background

In UPROM methodology, estimating the size of the system to be automated by means of business process models and analysis models plays an important role. The details of UPROM on size estimation procedures are provided in section 3.3.2. As explained there, the connections between functions and entities on FADs follow the pattern of basic CRUDL operations which are basic functions on persistent storage. This way of expressing functionality of the system both provide the descriptions of the services the system must provide, and it also makes it possible to map the services to basic data movements of COSMIC size measurement method. The mapping between the basic CRUDL operations and COSMIC data movements and additional rules are explained in section 3.3.2. To justify this mapping, before using the mapping and estimating the size in case studies, we decided to conduct a case study specific to size measurement.

In Software Management Research Group [204], we had requirements and COSMIC size measurement data defined for three different simple business application systems. Our group utilized these definitions for different experiments, so both the requirements and size measurement results were peer reviewed many times [205], [206]. These system definitions were for simple business applications. These applications were not designed to include long processes, but rather described basic operations conducted on the data. To evaluate how well UPROM performs on size estimation, we used these system definitions to develop simple business process models, and compare the results of the UPROM size estimation with measured COSMIC FP values.

We need to consider that these three systems which are objects of this case study describe the data movements in detail so that the size for the applications are identified precisely. UPROM describes an analysis methodology in business domain and from user perspective. Thus, UPROM is not used to express requirements in this much detail, as the aim is to develop user level requirements. However, COSMIC utilizes software requirements as input to calculate the measured size precisely. So, these cases do not have perfect correspondence with the cases where UPROM is used in real life. However, they provide a good opportunity for us to evaluate UPROM FSE method for detailed cases. The discussions on the results and evaluation of the applicability are provided below.

Paparazzi Information System

This is an application for managing celebrity information, keep relationships between celebrities, and record the catches of celebrities in specific places. The system definition and requirements document of the system is provided in Appendix A of the technical report providing UPROM case study results [207].

Veterinary Record System

The veterinary application is used to record pets, their owners and the applied vaccines. The system definition and requirements document of the system is provided in Appendix F of the technical report providing UPROM case study results [207].

Movie Manager System

Movie manager system keeps the information on movies together with its directors, producers, writers and actors/actresses. The enquiries and listings are also conducted for the movies. System definition and requirements document of the system is provided in Appendix K of the technical report providing UPROM case study results [207].

4.2.2. Objective

In order to answer the general research questions, this case study focused on integrating BPMod and functional size estimation practices. Thus, to contribute to answering the research questions (provided in 4.1.2), the research objective of this case study was to examine if UPROM FSE method can be used for estimating the size of three business application systems from business process models and evaluate if "reasonably accurate" size estimation can be achieved by means of UPROM FSE method compared to the measured size by COSMIC method.

4.2.3. Case Study Plan

The objective of this case study set was to use UPROM FSE procedures to estimate the size of applications that are already measured with COSMIC FSM, and justify our method for size estimation and provide a discussion on the differences. The results were planned to be utilized as a basis for discussions for the rest of the case studies.

The steps for each of the cases were as follows:

1. Develop business process models (EPC, FTD and VC) and analysis diagrams (FAD and ERD) for the system utilizing the system definitions and requirements.
2. Generate size estimation results by using UPROM FSE method.
3. Compare the previously measured size in COSMIC FP with the estimated value and evaluate the deviation.
4. Compare COSMIC FP and UPROM size measurement FP values on a functional basis and discuss the results.

4.2.4. Sources of Evidence

The objects of this case study are three business applications: Veterinary Record System, Paparazzi Information System and Movie Manager System. The subject of this study from which data is collected is the external expert. The sources of evidences to be collected during the conduct of the case study and to be analyzed to evaluate the results are listed below.

Documentation: This case study set was conducted based on existing definitions and results. Two types of existing documentation were utilized in each of the studies:

- A description and functional software requirements of the system to be measured
- Solution sheet developed according to COSMIC method manual [125]

All of these three systems were defined, measured, peer reviewed and results were utilized in different studies of our research group [205], [206]. Therefore, system definitions and size measurement results were reliable.

Physical Artifacts, Participant Observation and Related Outputs: We applied UPROM methodology on the given systems to make an estimation of COSMIC FP size of those systems. Therefore, we developed the following artifacts as an output:

- Business process models and analysis diagrams (EPC, FTD and FAD type diagrams)
- Generated size estimation report

As we are the ones that both developed the methodology and applied the results for the cases, we needed to evaluate if we applied the methodology properly to assure reliability and construct validity. For this purpose, a COSMIC certified measurement expert reviewed the outputs to assess if they are developed conforming to the UPROM methodology. Necessary updates were conducted according to review results.

- Participant Observations

As the appliers of the UPROM methodology, we observed and compared the size measurement results of COSMIC method and size estimation results of UPROM methodology. We reported the observed results of comparison and evaluations in the following sections. We used these data to analyze the results of the case study.

In this case study, we utilized metrics collection (such as estimated software size and previously measured size) for quantitative analysis, and observations for qualitative analysis of the results.

4.2.5. Conduct of the Case Study

For all of the three systems, business process models were developed using FTD and EPC and detailed analysis was conducted using FAD. During modeling, the aim was to reflect the system definition and requirements given for those systems by means of business process models. After the completion of the models, size measurement reports were generated by the tool and the results were analyzed as described in the following section.

As these three cases were for the systems that do not require long processes but describe small applications with basic functionality, we wanted to be sure if we were able to reflect system definitions and requirements in our models. Also, we had another concern for the reliability of estimation conducted by UPROM tool. We needed to ensure that UPROM tool generates size estimation values as specified by the methodology. To overcome these concerns, we asked a size measurement expert to examine the models and size estimation results for one of the models (Veterinary Record System is chosen at random) and answer the following questions:

- Do you think the business process models (together with FAD diagrams) are in conformance with the system definition and requirements given for the system?
- Do you think the operations on entities depicted on FAD diagrams are in conformance with the requirements statements?
- Please manually write size estimation results using FAD diagrams following UPROM size estimation methodology. Do your results match with the size estimation report generated automatically by UPROM?

The answers by the expert were positive. She stated that “the process model fits with the requirements properly” and “by using EPCs and FADs, one can easily understand the related requirements”. She also stated that the operations on FAD diagrams and requirements statements were in harmony. Also, the size estimation results that she developed manually matched with the generated report results. For the rest of the studies, we accept that UPROM size estimation generation algorithm complies with UPROM size estimation method rules.

Paparazzi Information System

The COSMIC size measurement for the Paparazzi Information System is provided in Appendix B; business process models report is provided in Appendix C; analysis diagrams report is provided in Appendix D; UPROM size estimation result report is provided in Appendix E of the technical report [207].

Veterinary Record System

The COSMIC size measurement for Veterinary Record System is provided in Appendix G; business process models report is provided in Appendix H; analysis diagrams report is provided in Appendix I; UPROM size measurement result report is provided in Appendix J of the technical report [207].

Movie Manager System

The COSMIC size measurement for Movie Manager System is provided in Appendix L; business process models report is provided in Appendix M; analysis diagrams report is provided in Appendix N; UPROM size measurement results report is provided in Appendix O of the technical report [207].

4.2.6. Data Collection and Analysis

Initial functional processes and size measurement comparisons are explained below. The results were utilized to refine the estimation rules. Updates results are provided later.

Initial UPROM FSE Results

Paparazzi Information System

- *Create Celebrity:* With the “create” operation of just one entity, the COSMIC size and UPROM estimated size matches.
- *Update Celebrity:* In COSMIC measurement, the update operation is handled in two steps: Retrieve Celebrity and Update Celebrity. In UPROM, to cover different cases in one operation type, we show it in one operation: Change. As a result, *1 E and two X data movements are missing* from UPROM measurement. However, considering the COSMIC Business Application Manual, it is a matter of assumption to consider these as two separate functional processes or just as one.

- *Delete Celebrity:* Together with the movements of R due to the business rule of the function, COSMIC size measurement and UPROM matches each other.
- *Search (Query) Celebrity:* List is utilized in UPROM for searching and listing the results of an entity. As this search operation is not related to any other entity, the listing is just for Celebrity, and it matches with COSMIC size measurement.
- *Add Catch:* Catch is an entity which utilizes other objects of interests (namely Celebrity and Place) as an attribute. Celebrity and place are listed in a drop-down list and selected as an attribute of Catch entity. Thus, in UPROM, in addition to showing "Catch" is created, Place and Celebrity are listed. However, as additional Entry movements are not necessary for one functional process, UPROM adds two extra "Entry" data movements with respect to COSMIC size measurement results.
- *Start Relationship:* During the creation of the Relationship entity, Celebrity is listed to be assigned as an attribute of Relationship. Also Relationship is listed to see if the celebrities are involved in other relationships. So, two List data movements are used in addition to creation on FAD. The same problem occurs with the previous functional process, 2 extra Entry data movements are estimated by UPROM.
- *Add Place:* Only the Place entity is created and the movements of UPROM and COSMIC size measurement match.
- *Delete Place:* As in the previous cases (Add Catch and Start Relationship), because of the operation (Catch) that is not conducted independently but related to Delete Place operation, extra one Entry movement is added on UPROM size measurement.

Veterinary Record System

- *Add Owner:* The Owner entity is created, and no other entity is related. UPROM size measurement and COSMIC results match.
- *Add Pet:* During the addition of Pet Entity, the related OOI is Owner, as it is listed and selected as an attribute of Pet. Because of this list operation, additional E data movement is added to UPROM size measurement.
- *Record (add) Consultation:* UPROM size measurement approach guides us to define a creation operation for Consultancy, and provide a list of Pets and Vaccines and add them as attributes of Consultation Entity. In COSMIC approach, a special entity named Pet-Vaccine is defined as OOI and E and W data movements are added for these. This OOI is defined because this requirement needs to list a special table: the list of the vaccines grouped by the type of pets. Although in UPROM we indicate this just as operation on Pet and Vaccine, the same result is reached as the intention is the same. Because of list operations, there are already two excess E data movements in UPROM size measurement. In total, two E data movements are excess and one X movement is missing in UPROM results.
- *List Pets:* As a single List operation on Pet entity, COSMIC and UPROM size measurement results match each other.

- *Update Pet:* In the same manner as "Update Celebrity" functional process in Paparazzi Information System, 1 E and 1 X data movements are missing from UPROM measurement.
- *Delete Pet:* The operation of Delete Pet has the same size in UPROM with COSMIC measurement. Additionally, for deletion of pet, consultation and vaccine information for the related pet is read and deleted. In UPROM methodology, as these deletion operations are not processed as embedded to the main deletion operation, 2 extra Entry movements are added to the measurement.
- *Search (query) Pet:* The requirement indicates that the vaccine and owner entities for the searched pets are also found and shown to the user. So, in UPROM methodology, we add 3 List operations, which add 2 extra Entry movements similar to the cases discussed above.

Movie Manager System

- *Add Person:* The Person entity is created, and no other entity is related. UPROM size measurement and COSMIC results match.
- *List Person:* As a single List operation on Person entity, COSMIC and UPROM size measurement results match each other.
- *Update Person Info:* In the same manner as "Update Celebrity" functional process in Paparazzi Information System, 1 E and one X data movements are missing from UPROM measurement.
- *Search Person:* As a single List operation on Person entity, COSMIC and UPROM size measurement results match each other.
- *View Person:* In this function, for the selected person, the information is viewed. The related entities include Director, Producer, Writer, Actor and Movie. In UPROM, view operation is modeled for all of these entities, and as the same reason explained before, 5 Entry movements are excess.
- *Delete Person:* During the deletion operation for Person Entity, the other related Entities, director, producer, writer and actor are also Read, and Producer, Writer and Actor are deleted. So, 4 Delete and 4 Read operations are modeled for UPROM size measurement. 3 of the Entry data movements are excess because of the related entities, and one X movement is missing.
- *Add Movie:* In UPROM methodology, we observe that the entity Movie is created during this functional process; and director, producer, writer and actor entities are listed so that they can be selected and added to Movie as attributes. As a result, one Create operation and 4 List operations are conducted.
  But if we were on a more detailed level of software analysis and developing a full E-R model, we would see that we could make up an M to N relationship between Movie and Person, and express each role (director, writer etc.) with a relationship. In this case, each of these M-to-N relationships would have their own table and they should have Entry and Write data movements of their own during the creation of Movie entity. Because of this, UPROM methodology estimates 5 W data movements less for this functional process (and 4 E movements will be less when Entry data movements are removed).

- *Search and View Movie:* As a single List operation on Person entity, COSMIC and UPROM size measurement results match each other.
- *View Movie:* In this function, together with the view operation of function, for the selected movie, the information is viewed. The related entities include Director, Producer, Writer, Actor and Person. In UPROM, view operation is modeled for all of these entities, and as the same reason explained before, 4 Entry movements are excess.
- *Delete Movie:* During the deletion operation for Movie Entity, the other related entities, Person, Director, Producer, Writer and Actor are also read, and Person, Director, Producer, Writer and Actor are deleted. So, 5 Delete and 5 Read operations are modeled for UPROM size measurement. 3 of the Entry data movements are excess because of the related entity operations.

Initial size estimation results in FPs for the three cases are summarized in below.

Table 7 Initial UPROM FSE results for case study set 1

	Paparazzi Information System	**Veterinary Record System**	**Movie Manager System**	**Average**
COSMIC FP	36 FP	37 FP	85 FP	53
UPROM FP	42 FP	44 FP	92 FP	59
Deviation of size	16,6%	%18,9	%8,2	14,5%

Considering the overall results of the three case studies, we observed that the most important cause for the diversion of UPROM size estimation results from COSMIC size measurement values is the introduction of additional Entry data movements. This problem took place when some operation (like list, change, delete, view) was conducted on an entity and other operations were added on FAD for the related entities because of this main operation. In this situation, the operations on related entities were not triggered by an external motivation, but triggered implicitly because of the operation on the main entity. Because of this, these entities did not need any Entry data movements, as they were not initiated by means of a data entry or external trigger. The functions in the case studies explained above where this situation occurred are listed below:

- Paparazzi Information System
    - Add Catch (2 extra Entry data movements)
    - Start Relationship (2 extra Entry data movements)
    - Delete Place (1 extra Entry data movement)
- Veterinary Record System
    - Add Pet (1 extra Entry data movement)
    - Record Consultation (2 extra Entry data movements)
    - Update Pet (1 extra Entry data movement)
    - Delete Pet (2 extra Entry data movements)
    - Search Pet (2 extra Entry data movements)
- Movie Manager System
    - Search and View Person (4 extra Entry data movements)

- Delete Person (3 extra Entry data movements)
- View Movie (5 extra Entry data movements)
- Delete Movie (3 extra Entry data movements)

To overcome this problem, we omitted the additional Entry data movements by utilizing ERD of the system. If, on an FAD, there were entities that are "related" with the main entity and an operation which involves an Entry data movement is applied for these entities; we removed these Entry data movements from the size measurement. The details of this rule are explained in section 3.3.2. For movie manager system, this brought a negative effect on Add Movie functional process. 5 Entry movements were removed, however, as explained above, these entries were required for separate data tables for person types.

When this rule was applied, updated UPROM FP size measurement results and deviation from COSMIC results are updated as seen in Table 8. In the overall, the average deviation decreased from 14,5% to 3,6%.

Table 8 Updates UPROM FSE results for case study set 1

	Paparazzi Information System	**Veterinary Record System**	**Movie Manager System**	**Average (of absolutes)**
COSMIC FP	36 FP	37 FP	85 FP	53
Updated UPROM FP	38 FP	36 FP	83 FP	51
Deviation of size	5,6%	-2,7%	-2,4%	3,6%

Considering that early FSE conducted by UPROM is based on immature data compared to software requirements in later phases, we don't expect the estimated size to match with measured functional sizes of these systems. Even when a system with mature requirements are measured by trained measurers using COSMIC method, measured size can deviate largely because of individual interpretations and assumptions. Experimental results show that more than 20% deviation is observed in most of the measurements caused by different interpretations [205], [206], [208]. The study of Santillo and Meli suggested 10% deviation benchmark [20]. Adding up two aspects of deviation, we accept that up to 30% deviation is reasonably accurate for early FSE method.

The COSMIC functional size of these three systems measured by UPROM size measurement methodology deviated from the actual COSMIC FP values by less than 10%, and by about 3,6% on the average. Considering that the cases in this study had mature software requirements specifications, lower deviation was expected. Estimated values for the three cases individually were below 10% deviation benchmark. Thus we conclude that UPROM provides reasonably accurate estimation.

As a result of the case studies, we answer the research questions in the following way:

- CSQ1: We could use UPROM size measurement methodology to provide an early estimate for the size of the system to be automated in Function Points.

- CSQ2: We conclude that our results were "reasonably accurate", as we observed that the deviations were within the benchmark limits. We keep in mind that this is an early size estimation method, and all details of software have not yet evolved. Considering the maturity of the business process modeling and analysis study and the complexity of the system, we anticipate and find it acceptable that the deviation may be higher as all the complexity of the system may not be reflected to process models in this stage.

4.3. Case Study Set 2 - e-LegalEntity Company Central Registration System and e-Trademark Central Trademark Registration System

In this section the second case study of the study is explained. First, the background information for the two projects is explained. Case study questions are defined. Then, the plan for the case study conduct is given. Sources of evidences are identified. In the last section, conduct of the case study, data collection and analysis results are provided.

4.3.1. Background

Company Central Registration (e-LegalEntity) project aimed to automate the establishment of new legal entities and management of changes in their data through their life time. Another goal of the project was to develop the system so that different types of legal entities which are registered separately at the moment is gathered under one database and given a unique id. In project's scope, only the processes related to companies were to be automated. The rest of the legal entity types were to be registered to the system by simple registration processes, which required basic processes to be conducted.

Within the same project, one more system was to be developed: Trademark Registration System (e-Trademark). By means of this system, all the trademark applications, approvals and related workflows were to be automated by an online application. e-Trademark project's scope included operations on patent and geographical indications together with trademarks. As the processes are very similar for three of them, it is decided to analyze the trademark processes in detail, and reference to it for the patents and geographical indicators.

The project was subcontracted to our analyst group by the contracting organization. Our group consisted of three analysts one of which is the researcher of this study. The three experts from the contracting organization also worked as the analysts in the project; they conducted peer reviews on the artifacts, changed, updated and approved them together with our group. All the documents were also presented to domain experts which were composed of officials from different entities and their consent was necessary to finalize the studies. Both e-LegalEntity and e-Trademark system studies were conducted by the same analyst group, but different domain experts (one for each) were responsible for them. To ensure that all the stakeholders were in common terms, a one day training was provided both for the analysts of the contracting organization and the domain experts.

The training topics included basics of business process modeling, modeling with EPC and brief explanation on the UPROM methodology and usage of the tool.

The contractor required that the analysis of the current situation is conducted, the user requirements of the system to be automated is analyzed and defined, the development of the software is planned and the processes to be performed by means of the software is disseminated to the stakeholders as written process definition documents. Then, the development of the software was subcontracted with the technical contract that is developed by using the developed user requirements.

The initial phase of the projects covered the analysis of as-is business processes, definition of to-be processes, preparation of technical contract document, estimating software size and supplying documentation for end users to understand the designed to-be system and utilize in the operational phase of the software. We identified these projects as the object of this case study and followed UPROM methodology to analyze business processes and user requirements and generate the artifacts. These activities were conducted separately both for e-LegalEntity and e-Trademark systems. Thus, this is a new implementation type case study, as UPROM was followed from scratch to conduct the project activities and deliver the outputs.

As the scope of this case study, in e-LegalEntity project, we only focused on the company processes within the project. The rest of the legal entities only follow basic processes to be recorded to the system. The requirements analysis for those entities focused on data to be collected rather than automation of processes. So we scoped out the processes related to other legal entities from the case study.

Another important aspect in the scope of this study is that, although the as-is models are analyzed, the business process models were only developed for the to-be models. Thus, although the constraints and data requirements and interfaces between the systems remained the same, the flow of the activities was completely updated in the to-be models.

The related stakeholders for the company processes were Company Registrar's Office, the natural entities having the intention to establish a company and persons in authority for the companies. 20 personnel responsible for the related processes were working in the Company Registrar's Office. The total number of companies registered was more than 30.000. The basic processes in the scope are as follows:

- Establishment of a new company
- Update of existing company information
- Approval of company applications
- Listing of companies
- Closure and clearance of companies

The related stakeholders for the trademark processes were Industrial Properties Office, the natural and legal entities that require to register a new trademark and that already have trademarks. 2 personnel were responsible in the office for the related processes.

Over 10.000 trademarks were already registered. The basic processes in the scope are as follows:

- Registration of a new trademark
- Regular update of a registered trademark
- Update of trademark owner
- Update of existing trademark information
- Update of franchising information of a trademark
- Update of trademark deputy

4.3.2. Objective

In order to answer the general research questions, this case study focused on integrating BPMod, requirements analysis, functional size estimation and process documentation practices. Thus, to contribute to answering the research questions (provided in 4.1.2), the research objective of this case study was to examine, for e-LegalEntity and e-Trademark projects, if UPROM can be used to analyze and model processes, generate process documentation, develop user requirements and estimate the size of the software to be automated all together and based on business process models. Furthermore, it is aimed to evaluate if the related artifacts better than the ones developed with traditional methods in terms of consistency, completeness, maintainability and traceability.

4.3.3. Case Study Plan

The objective of this case study set was to apply UPROM to develop the artifacts required by the project, collect and analyze the data and answer the research questions. Planned list of activities to conduct both of the case studies in this case study set, the data to be collected, data collection methods and the analysis methods are listed in Table 9. This table, although prepared for this case, are also utilized for the rest of the cases.

Table 9 Case study plan for e-LegalEntity and e-Trademark Projects

No	Activity	Inputs	Data to be collected	Data Collection Method	Analysis
1	Conduct descriptive business process analysis Add process meta-data, description, terminology, improvements	As-is process definitions Decisions on to-be processes	# processes # nodes, connectors, functions and arcs # diagram types	Metric collection	Analyze the coverage Analyze the contribution of the guidelines Identify improvements suggested
			Evaluation of guidelines, expressiveness of models, project structure	Observation Interview	
2	Conduct function allocation analysis	As-is process definitions Decisions on to-be processes			

		Information from domain experts Contract			
3	Generate requirements document Identify the functional requirements not included Analyze if generated requirements conform to UPROM procedures.	Business process models (FTD, EPC, FAD, ERD) Generated requirements document	# total functional requirements # functional requirements generated and manually added # generated functional requirements from constraints % of generated requirements	Metric collection	Analyze the coverage of generated requirements Evaluate completeness, consistency, maintainability and traceability.
4	Generate FSE report Measure a sample by COSMIC	Business process models	FP size measured by UPROM FP size measured by measurement expert	Observation Interview	Evaluate accuracy of FP estimates Evaluate team's perception
5	Generate process definition document, business glossary, process improvement list Generate process KPI list (N/A for 2nd case) Analyze conformance to UPROM procedures	Definitions from existing documentation and domain experts' knowledge Business process models	%of descriptions # and % business glossary entries # improvements # KPIs (N/A for 2nd case)	Metric collection	Analyze the content coverage and users' perception Analyze the structure and usability Analyze maintainability and traceability
			Answers and observations for content coverage, understanding and benefits Evaluation of maintainability	Observation Interview	
6	Analyze all outputs generated as a whole Evaluate benefits brought by integrating activities	All outputs and analysis results from previous activities	Errors, deficiencies, overlaps, inconsistencies found in the case study The list of good practices, positive and negative findings		Analyze the outputs as a whole to answer the research questions Evaluate the validity and reliability

4.3.4. Sources of Evidence

The objects of this case study are two projects within the e-Government program: e-LegalEntity and e-Trademark. The subjects of this study from which data is collected are the analysts from the contracting organization, analysts from contractor company and end users with which interviews are conducted and an external expert. The sources of evidences to be collected during this case study is listed under "Data to be collected" column of Table 9. Sources of evidences can be summarized as follows:

Documentation: Existing documentation to understand the as-is processes, like laws, can be listed under this type of source of evidence.

Interviews: This was the most important data collected in this case study. By means of the interviews, we were able to gather ideas of different types of stakeholders which were totally immersed in the project activities. Interview questions utilized for e-LegalEntity and e-Trademark projects can be found in Appendix P of the technical report [207].

Direct Observations: We found the chance to make direct observations on the case in its natural settings. For example, reviews, feedbacks and updates from both the contracting organization analysts and domain experts were indeed observed in the projects' regular activities without any interruption for the case study and while we, as the case study performer, were not involved in the activities. Direct observations were utilized while we analyze and evaluate the results in the following section.

Participant Observation: Working as the analyst in the project, many observations were conducted as participants in the study, by means of which we had a deep understanding of the implementations and collect feedbacks. The results are also utilized to analyze and evaluate results in the following section.

In this case study, we utilized metrics collection (such as percentage of generated requirements in the technical contract document) for quantitative analysis, and observations and interviews for qualitative analysis of the results.

4.3.5. Conduct of the Case Study, Data Collection and Analysis

Conducting Descriptive Business Process and Function Allocation Analysis

We started conducting the case study by developing business process models for e-LegalEntity and e-Trademark projects following UPROM. At first step, business process models were developed (FTD, EPC and OC diagrams) as mentioned in the ***Activity 1*** of the case study plan (Table 9). As an input, documents for the as-is processes and decisions with the contracting organization analysts and domain experts on the to-be processes (derived from the meetings) were utilized.

The business process models including FT, EPC and OC diagrams for e-LegalEntity and e-Trademark are provided in Appendix R and Appendix Y of the technical report respectively [207]. The detailed analysis models including FAD and ERD diagrams for e-LegalEntity and e-Trademark are provided in Appendix R and Appendix Z of the technical report respectively [207]. An example EPC can be seen in Figure 43, an example FAD can be seen in Figure 44 and an example ERD can be seen in Figure 45. All examples are from e-Trademark project.

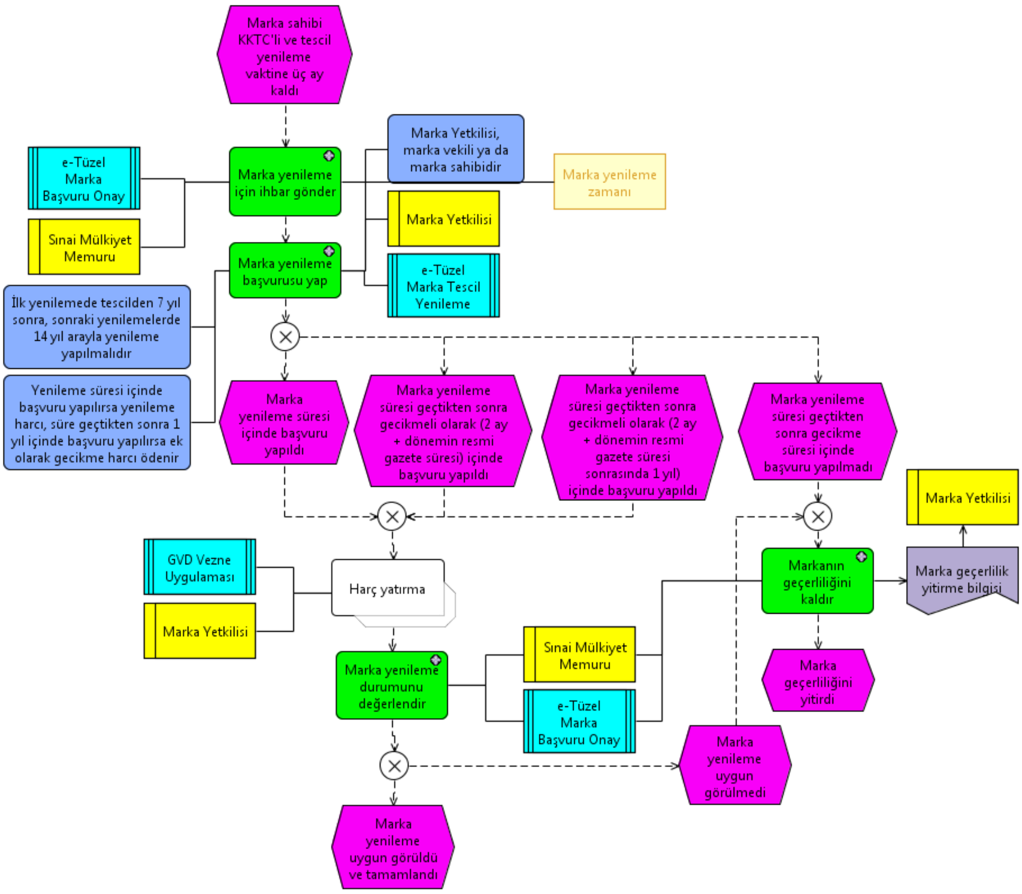


Figure 43 An example EPC from e-Trademark project

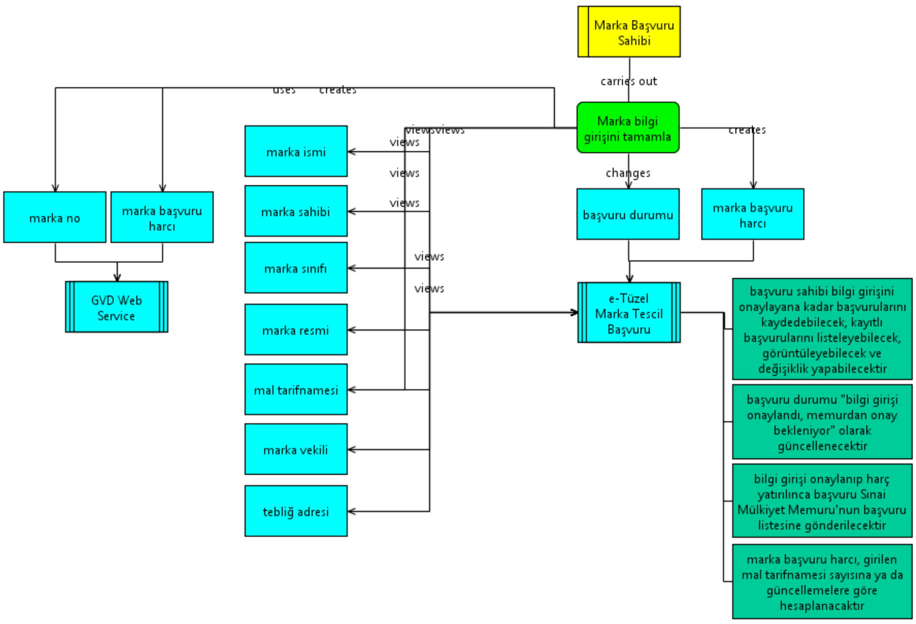


Figure 44 An example FAD from e-Trademark project

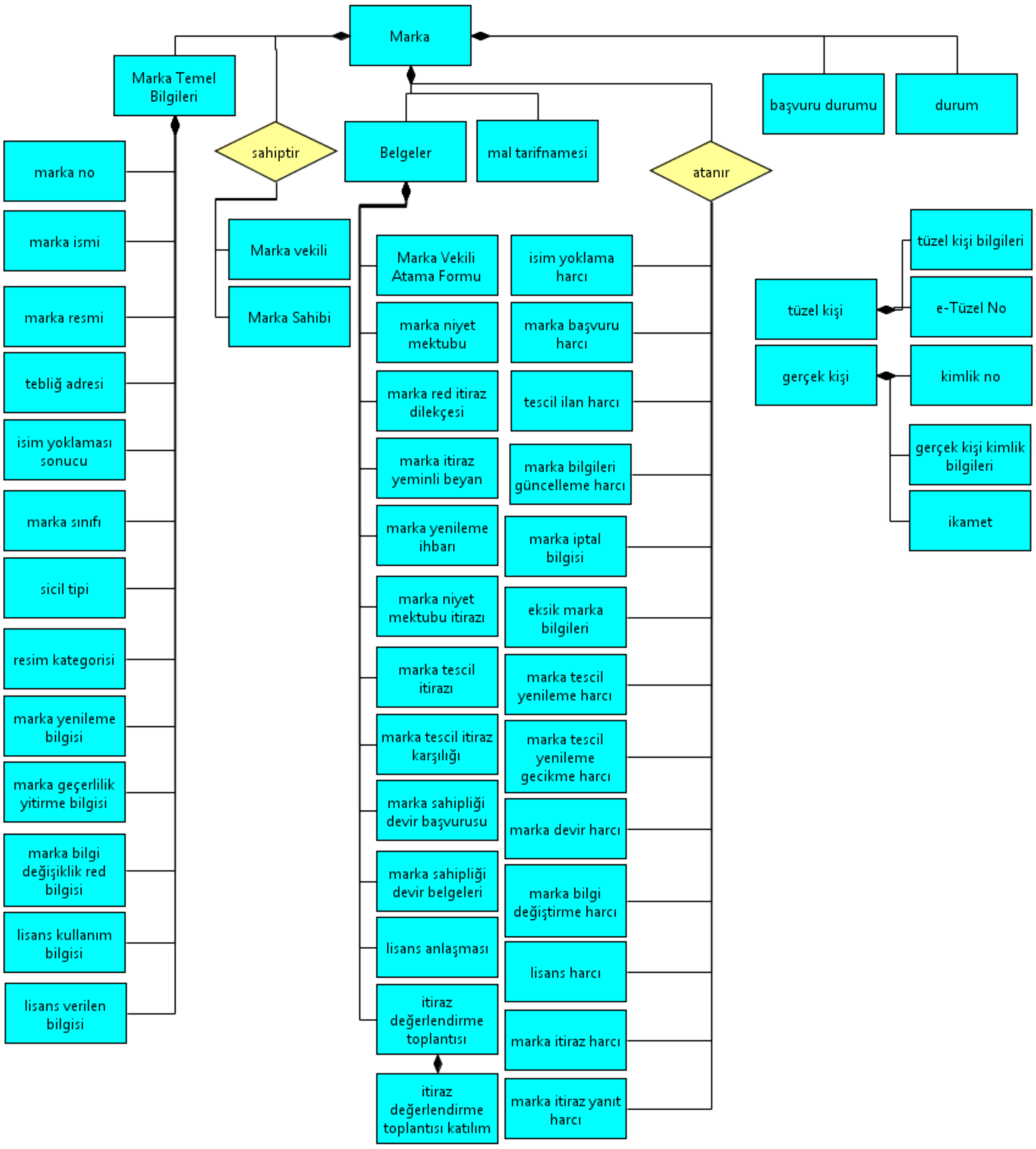


Figure 45 An example ERD from e-Trademark project

The structure of the project under the main project "eTuzel" (standing for e-LegalEntity) is shown in Figure 46. FADs are not shown for the sake of readability, but they are placed under the same folder with their related EPC diagram from which it is referenced as a sub-diagram. As seen in the figure, under the main project folder, there are three models:

- eTuzel.erd: Provides the high level entities and their relations in the system (shown in Appendix R of the technical report [207]). It covers all other legal entities that are within the scope of the project. The details of the entity "Şirket" (company) were provided in the ER sub-diagram named as Sirket.erd (shown in Appendix S of the technical report [207]).
- eTuzel.ftd: the process map file for the project. All the process models in one level lower are linked on this process map. There are functions referencing the

processes of all legal entities within the scope of the project. These do not have links to their detailed process models, but just shown to provide a comprehensive list of entities. One lower level process map providing details for company legal entity is modeled as “Sirket.ftd” process model.

- eTuzel.oc: Organization chart that includes all organizational elements and their relations within the project.

Under the main project folder, there are subfolders for the highest level processes in the project. The focus for our case study was the folder “01-Sirket”, containing the processes for the company. Our focus was on company legal entity, which has its own process map, “Sirket.ftd”, which also provides reference to lower level EPC diagrams.

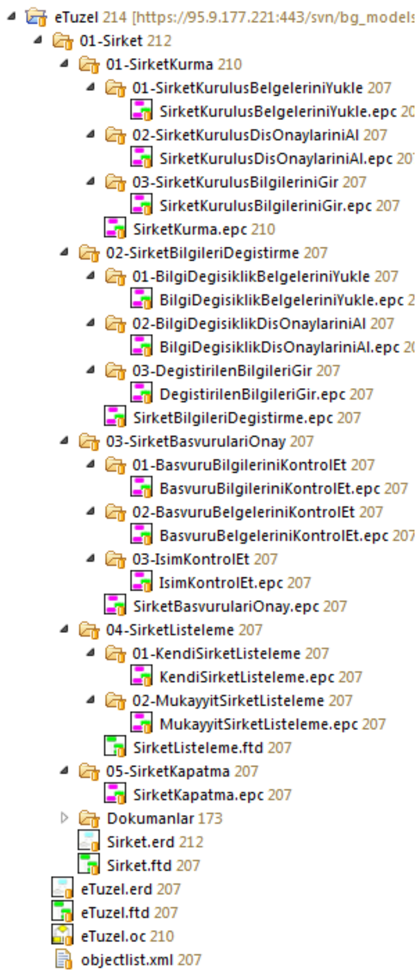


Figure 46 Modeling project structure of the e-LegalEntity

The rest of the subfolders were organized in number to “logically” order the diagrams so that the models are easily reachable. The names of the EPC models in order and in hierarchy are as follows:

- Establish new company
    - Enter company information

  - Add company establishment documents
  - Obtain external approvals for the establishment
- Update the data of existing company
  - Enter updated information
  - Add updated documents for data update
  - Obtain external approvals for data update
- Approve company applications
  - Check proposed company name
  - Check company application information
  - Check company application documents
- List companies
  - List own companies
  - List companies as the official in charge
- Closure of the company

The structure of the project under the main project "MarkaPatent" (standing for e-Trademark project) is shown in Figure 47.

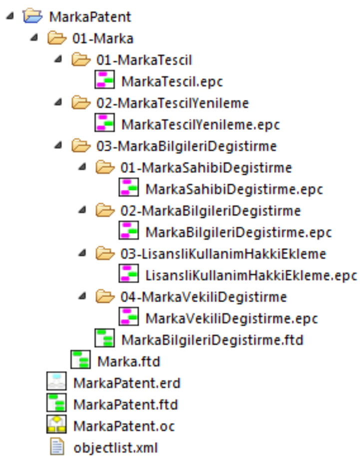


Figure 47 Modeling project structure of the e-Trademark

As seen in the figure, under the main project folder, there are three models:

- MarkaPatent.erd: Provided the high level entities and their relations in the system (shown in Appendix Y of the technical report [207]). It covers all of the entities that are within the scope of the project; trademark, patent and geographical indicators. However, only the details of the entity "Marka" (trademark), which is the only entity modeled in detail are provided in this ERD.
- MarkaPatent.ftd: the process map file for the project. All the process models in one level lower are linked on this process map (in Appendix Z of the technical report [207]). There are functions referencing all main entities (trademark, patent and geographical indicator) within the scope of the project. These do not have links to their detailed process models, but just shown to provide a

comprehensive list of entities. One lower level process map providing details for trademark entity is modeled as “Marka.ftd” process model.
- MarkaPatent.oc: Organization chart that includes all organizational elements and their relations within the project.

Under the main project folder, there are subfolders for the highest level processes in the project. The focus for our case study is the folder “01-Marka”, containing the processes for the trademark. Our focus is on trademark entity, which has its own process map, “Marka.ftd”, which also provides reference to lower level EPC diagrams.

As in the case of e-LegalEntity project, the subfolders in this project are organized in number to “logically” order the diagrams so that the models are easily reachable. The names of the EPC models in order and in hierarchy are as follows:

- Register new trademark
- Update the registration of an existing trademark
- Update trademark information
  - Change trademark’s owner
  - Update existing trademark information
  - Update franchising for the trademark
  - Update trademark’s deputy

In the modeling of both e-LegalEntity and e-Trademark projects, the following symbols are used:

- **EPC:** Event, Function, Process Interface, Application, Organizational Elements (Organizational Unit, Position, External Person), Document, Business Rule, Connectors (And, Or, XOR)
- **FTD:** Function
- **FAD:** Function, Entity, Application, Organizational Elements (Organizational Unit, Position, External Person), Constraint
- **ERD**: Entity, Relationship
- **OC**: Organizational Unit, Position, External Person

EPC diagrams were modeled in detail, providing alternative paths. The live validation functionality was utilized and EPC models were developed free of semantic errors for both e-LegalEntity and e-Trademark projects other than two:

In both “Establish new company” and “Update the data of existing company” processes, there is one error:

“Because of an (X)-OR split before the AND-join, the control flow might not reach all incoming arcs of the AND-join.”

We are aware of this possibility and accept this as there are alternative exits for the process. 10 out of 15 EPC diagrams for e-LegalEntity project and 5 out of 6 diagrams for e-Trademark project have more than one start or end event. Because of this, these diagrams are not sound. But as we discuss in the methodology, because our focus is on business process analysis rather than formal analysis of the processes, multiple start and end events supported analysis activities.

FADs were developed for most of the functions on the EPC models, as most functions were envisioned to be automated. All of the entities utilized in FADs were placed on ER diagrams so that entities and relations between them can also be observed as a whole and size estimation rules can be applied accurately.

The structure of the project was established as explained in UPROM methodology. Subfolder structure corresponded to the sub-diagram hierarchy developed in the diagrams. Each of the business process diagrams (FT or EPC) were placed under a subfolder. The name of the folder and the name of the EPC/FTD diagram matched each other (the folder may have an extended name like a number added). On UPROM tool, the project structure was validated successfully. For any problem for nonconformance to the structure (like naming the folders wrongly), the validation did not permit any reporting to be generated.

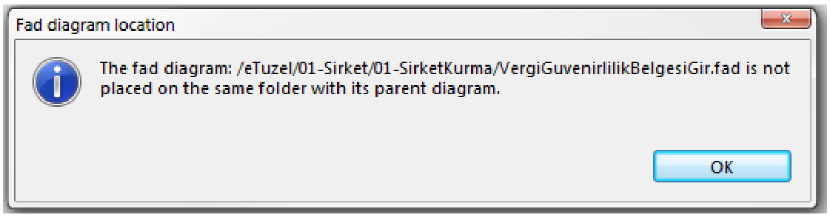


Figure 48 Warning for misplacing the FAD

When the validation was conducted, there was only one type of error as shown in Figure 48. The same error was given for 5 different FAD diagrams in e-LegalEntity project and 2 different FAD diagrams in e-Trademark project. The reason for this is that, same FADs are utilized for the same functions of both “Establish New Company” and “Update the data of existing company” processes in e-LegalEntity project. The same applies for “Register trademark”, “Change Trademark Owner” and “Update trademark information” processes in e-Trademark project. The FADs were originally placed under the folders of the first processes and the others are just references to those, as the same functions were relevant for the other processes, too. This error warned about the situation but did not create problems the structure of the project or prevent the generation of any reports.

Data Collection – Metrics

The metrics for the developed processes are shown in Table 10. Detailed metrics for EPCs in e-LegalEntity and e-Trademark projects are provided in Table 11 and Table 12.

Table 10 Metrics for e-LegalEntity and e-Trademark business process models

Metric	**e-LegalEntity**	**e-Trademark**	**Explanation**
Deepest level in the hierarchy	3	3	
# EPC diagrams	15	6	
# FTD diagrams	3	3	
# FAD diagrams	82	36	
# functions on EPCs	110	40	16 functions are not assigned FAD as they reference a sub-diagram or are not to be automated
# functions per EPCs	7,3	6,7	
#FADs per EPC	5,5	6	
# unique control flow nodes	328	122	Includes functions, connectors, process interfaces and events
# unique control flow nodes without event nodes	183	79	
Average number of total control flow nodes per EPC diagram (without events)	12,2	13,2	
Average number of connector nodes per EPC diagram	4	4,7	
Average number of arcs per EPC diagram	24,7	21,8	

Table 11 Detailed metrics for EPCs in e-LegalEntity

	Function	Connector	Prc.Int. (instance)	Control flow nodes w/o events	Event	Control flow nodes	App.	Org.El.	Inf.Car.	Bus.Rule	Total	# of arcs	Start Event	End Event
SirketKurma	14	11	5	30	20	50	3	2	10	2	67	56	1	4
SirketKurulusBilgileriniGir	11	2	0	13	4	17	1	1	0	3	22	23	1	1
SirketKurulusBelgeleriniYukle	11	2	0	13	9	22	1	7	14	0	44	30	1	1
SirketKurulusDisOnaylariniAl	6	3	0	9	6	15	1	1	0	0	17	18	1	2
SirketBilgileriDegistirme	11	11	5	27	21	48	3	2	10	2	65	51	2	4
DegistirilenBilgileriGir	12	3	0	15	13	28	1	1	0	2	32	36	2	1
BilgiDegisiklikBelgeleriniYukle	10	2	0	12	11	23	1	4	12	1	41	30	1	1
BilgiDegisiklikDisOnaylariniAl	5	3	0	8	7	15	1	1	0	0	17	17	1	2
SirketBasvurulariOnay	5	5	0	10	13	23	1	1	0	0	25	22	2	8
IsimKontrolEt	2	1	0	3	4	7	1	1	0	0	9	6	1	2
BasvuruBilgileriniKontrolEt	3	1	0	4	6	10	1	1	0	0	12	9	1	3
BasvuruBelgeleriniKontrolEt	3	1	0	4	6	10	1	1	0	0	12	9	1	3
KendiSirketListeleme	6	4	2	12	6	18	1	1	0	1	21	19	1	1
MukayyitSirketListeleme	4	2	1	7	4	11	1	1	0	0	13	10	1	1
SirketKapatma	7	9	0	16	15	31	1	2	0	4	38	35	2	2
Total	110	60	13	183	145	328	19	27	46	15	435	371		
Average	7,3	4,0	0,9	12,2	9,7	21,9	1,3	1,8	3,1	1,0	29,0	25		

Table 12 Detailed metrics for EPCs in e-Trademark

	Function	Connector	Prc.Int. (instance)	Control flow nodes w/o events	Event	Control flow nodes	App.	Org.El.	Inf.Car.	Bus.Rule	Total	# of arcs	Start Event	End Event
MarkaTescil	21	14	6	41	15	56	3	3	2	12	76	63	1	3
MarkaTescilYenileme	4	4	1	9	8	17	3	2	1	3	26	19	1	2
MarkaSahibiDegistirme	6	3	1	10	6	16	3	2	1	1	23	16	2	2
MarkaBilgileriDegistirme	4	3	1	8	6	14	3	2	0	1	20	14	1	2
LisansliKullanimHakkiEkleme	3	2	1	6	5	11	3	2	0	0	16	11	1	2
MarkaVekiliDegistirme	2	2	1	5	3	8	1	2	0	0	11	8	1	1
Total	40	28	11	79	43	122	16	13	4	17	172	131		
Average	6,7	4,7	1,8	13,2	7,2	20,3	2,7	2,2	0,7	2,8	28,7	22		

According to “7 Modeling Guidelines” [9] and related studies on which it is based on, the more elements a process model has, the more the error probability is. [175] suggests that, according to their experimental study, frequency of errors in a model reaches more than 50% if the following number of nodes are achieved:

- $S_N > 48$ (number of nodes)
- $S_c > 8$ (number of connectors)
- $S_F > 40$ (number of functions)
- $S_A > 40$ (number of arcs)

For e-LegalEntity and e-Trademark projects, the average values are provided in Table 10 and number of different types of nodes for each EPC diagram is provided in Table 11 and Table 12. We calculated the total number of nodes in this study as control flow nodes without events. As observed, average number of all nodes, function and connector nodes were less than the suggested values. Considering individual EPC diagrams, none of the diagrams exceeded the number of all nodes limit and 2 of them in e-LegalEntity and 1 of them in e-Trademark projects exceeded the number of connectors limit. Only one model in each of the projects exceeded the number of arcs limit. In the overall, we can say that our models complied with the 7PMG [9] for the size of the models (for G1, G2 and G7).

Usage of single start and end element is encouraged by the same guideline (G3). However, we need to utilize multiple start and end events to express different triggering functions and end states; and we allow usage of them if the rest of the flow is similar for multiple start and end events. In e-LegalEntity project, 4 processes had multiple start events. 9 processes had multiple end events over 15 processes, but 5 of them had the same events with the other 3 because they were sub-process or used as a process interface from the 3 processes. In e-Trademark project, 1 process had multiple start events, and 5 projects had multiple end events over 6 processes. Special focus was given to match start and end events in UPROM methodology so that the flow is smooth between the main diagram, sub-diagrams and process interfaces, as described in 3.2.2 of the methodology. We observed in this case study that the multiple start and end events enhanced expressiveness of the model and decreased error probability as total number of nodes and similar activities decreased by means of grouping the activities for multiple start and end events.

As suggested by [9] (G4), models shall be structured for splits and joins. In e-LegalEntity and e-Trademark projects, all splits and joins were paired apart from the ones used to indicate multiple close and end events, and the joins that indicate feedback returns (that were used to indicate some part of the process to be restarted because of a feedback in process activities).

Or routing element is suggested to be avoided (G5) to provide clear semantics of the models. We suggest conforming to these guide, replacing OR joins with alternatives as explained in UPROM methodology. However keeping in mind that we conduct analysis in the business level and sometimes need to indicate that not all or only one of the different paths need to be chosen. An example of this on "Enter updated information" process of e-LegalEntity project is as follows (Appendix R of the technical report [207]). During the conduct of a company's processes to update its data, the company may need to update some or all parts of its data. For that reason, all of the update activities were grouped between an OR split-join pair. It is necessary for us to express the capability of the process to enable conducting of update operations in parallel, either one, some or all of them. This was also the case when the process is to be automated. That's why we didn't avoid usage of OR connectors in these cases. If we were to join all update activities in into one to avoid the OR connection, we would cause that one activity to be very complex, deviating from the general granularity level of the functions. If we were to replace OR with AND connectors in this model and use XOR split-join pairs for each path between AND split and join, it would cause a complex and hard-to-understand view for 10 parallel flows from the viewpoint of the user. The similar case holds for the processes "Obtain external approvals for establishment", "Add updated documents for data update", "Obtain external approvals for data update" and "Enter updated information" processes. We paied importance for not using OR connections for multiple start and end events, as it causes ambiguity to decide when the process starts or ends.

The depth of the hierarchy in both projects were 3, which was enough to compose a hierarchical structure but not much to increase error probability and decrease understandability. The level of granularity for the functions was tried to be kept same throughout the projects. In some processes (like establishment of a company), separate functions were placed to define each sub-entity and related entities for the company. In some processes (like listing), only one function was placed to observe the company entity. This was because when the user intends to view the company, it is required that she can view all information on sub-entities. However, to balance this difference in the level of granularity, the view operation on the related entities were shown separately on FAD. All of the activities were labeled in an imperative way, as suggested by the guideline (G6) and UPROM methodology.

Thus, as mentioned in sub-proposition (a) to answer the research question 2 of the case study, the project conformed to the modeling guidelines.

Data Collection – Summary of Observations

Direct and participant observations to conduct the analysis specified in the case study plan for e-LegalEntity and e-Trademark projects are listed below. Observations for the two projects are provided together.

- Evaluation of the extent of information covered in current processes but not covered in process models:

The models were developed without any problems for this case study set. The symbol set was more than enough and only a limited number of symbols were used. We added descriptions for the functions to make further explanations on in what cases the functions were conducted and further controls or manual operations during their execution.

To our knowledge, we could capture the envisioned information in the process models. We did not need any further explanation or document to describe the processes. However, we needed two extra documents. One is a "Data Table" which describes the detailed attributes of each entity placed on ERDs. The other is "Data Interface Table", which listed the data to be provided to other systems by our system, and the data that needs to be obtained from external systems. These were information regarding to data and not processes, and the processes were used as input to prepare these documents. Considering this, we can say that process models were sufficient to express the process information, and was helpful to provide a starting point to further describe the entities.

- Evaluation of how the models helped to design the flow of to-be processes

Our group worked as process analysts in this study. Trying to group the activities as processes helped to understand the start and end of specific functionalities. For example, for the activities on the establishment of the company, the group was not sure how to organize the process, if it is necessary to diverge the official's approval activities and applications or not. Working on the processes, it turned out that the establishment should be handled as a whole, and approval activities are modeled separately and referenced as a process interface from the establishment process. The processes were modeled conforming to the meta-model of the notation as explained 3.1 of the methodology. Conformance to the meta-model is enforced by the tool.

The processes also helped to prepare the non-process documentation. The "Data Table" explained above were developed based on ERDs. The "Data Interface Table" was prepared by checking every function on processes and observing if it has an incoming or outgoing interface with any external system.

- Evaluation of the project structure

The project structure helped to organize the processes in an understandable and maintainable hierarchy and work in harmony as a group, as explained in the above bullet.

We were able to reach the models easily, update the files and folders without changing the previous structure and understand the processes.

Data Collection – Summary of Interviews:

The analysis of interview questions by different types of project stakeholders are provided below for each question. A general evaluation of the interview results and the overall findings are presented for each topic. Interview questions are provided in Appendix P of the technical report [207]. Interviews were conducted in a semi-structured way and the conversations were recorded with a voice recording device. The summary and highlights of the interviews for each question are provided in Appendix Q of the technical report [207].

- All interviewees conducted process analysis and definition activities before, but only three of them used BPMod notations like flowcharts.
- All of the interviewees agreed that specific characteristics of the notation like business rules, application systems, usage of consistent events for start and end of processes, sub-diagrams and process interfaces for modular processes, using the same object definitions for the whole project enhanced the expressiveness and understandability of the processes. Three of them specifically emphasized that these aspects enhanced the capability of them to reveal any problems or possible design alternatives in the system.
- The analysts (five of the interviewees) stated that defining the objects uniquely throughout the project and reflecting the updates for all of them is a critical property to conduct the process analysis. They also stated that the hierarchical structure enforced by the tool is helpful for organization during modeling and enhances understandability of the models. We could not get the idea of domain experts for this aspect as they didn’t use the tool but the reports to read the models.
- All of the interviewees indicated that usage of multiple start and end events do not decrease the understandability of the processes. Five of them specifically emphasized that they are a must for proper analysis of the system, and they can follow the relation between the models by means of them.
- All of the domain experts indicated that they would prefer to read process definitions in natural language previously, but after this study they would clearly like to examine process models. They mentioned that they would like to have models and written process descriptions together (this point will be detailed in process document generation part). During the analysis phase, both of the domain experts provided detailed feedbacks on the models, identified many additional features of the system and possible alternative paths which they didn’t describe in the first place. They stated that this situation was enabled by means of process models. It would be a lower possibility that they would reveal so many issues about the system without the process models. One of the analysts from our group indicated that he prefers to use the models alone, but documents can be helpful for different stakeholders.

- Analysts from the contracting company also indicated that although they did not prefer to conduct BPMod previously, they would like to use it for analysis for the following studies.
- For the question asked to the analysts, they indicated that the time for analyzing and designing processes in natural language or with BPMod would not differ too much. However, when it comes to maintain a process definition, that is, reflecting the effects of a change in the processes throughout the project, BPMod clearly takes less time. They mentioned that especially by means of the unique object property, they find the instances of the objects possible to be affected and consider the updates using this information.
- Analysts also responded that the meta-model usage enforced by the tool enabled better process analysis as they needed to comply with the predefined constraints on the symbols and conduct the analysis in a standard way. The experienced analyst indicated that the symbol and connection usage specific to our notation enhanced the analysis capabilities.

To sum up, evaluating the interviewees' responses, we can conclude the following.

All of the interviewees stated that business process models provided enough information for the related processes; especially the standard usage of symbols like application, business rules, organizational elements for all relevant functions and process structuring by means of sub-diagrams and process interfaces helped to describe the processes as a whole. However, as they also mentioned that they like to read detailed information on the activities from natural language documents, we understand that they needed to add detailed descriptions for the functions.

An analyst from our group indicated that he would like to clearly specify which events and other symbols are internal objects, which are related to external world. We utilize this idea as a potential future work.

We observed that many aspects of the methodology like unique object definition, predefined project structure, restricted meta-model usage, consistent usage of multiple start and end events support the analysts to design the processes. We observed that the analysts considered business process model usage for further process analysis activities although they preferred natural language usage previously, and domain experts requested business process models for better understanding of the processes.

Generating Requirements Document

As mentioned in Activity 2 of the case study plan, in this step requirements were generated for the system by means of UPROM tool's requirements generation function. The tool generated the requirements automatically from business process models as explained in the methodology with some changes to the form of the sentences. These changes were required by the project. We assured that the tool generates the requirements statements as defined in 3.3.1 of the methodology section. For this, we provided a researcher with the description of the requirements generation methodology, e-LegalEntity and e-Trademark process models, generation requirements specific to these projects (explained below) and the generated requirements. The researcher, who was not involved in project activities, analyzed the generated requirements to see if they conform to the methodology, and approved that they do.

The generated requirements document was analyzed by the project team and new requirements are added manually to develop the technical contract for the project. The added requirements were on general properties of the system and data interfaces.

The generated user requirements document can be seen in for e-LegalEntity, Appendix T and for e-Trademark, Appendix AA of the technical report [207]. Generated requirements are utilized in "technical contract documents". Excerpt from the generated requirements document can be seen in Figure 49.

1. Süreç adı: Sirket.ftd

Süreç adresi: eTuzel/01-Sirket

1.1. Süreç adı: SirketKurma.epc

Süreç adresi: eTuzel/01-Sirket/01-SirketKurma

1.1.1. REQ1. Şirket kuruluşu için başvurma işlemini Şirket Tescil Başvuru Sahibi yürütecektir.

1.1.2. REQ2. Bu işlem sırasında eTuzel Şirket Kurma Başvuru modülünde başvuru durumu ve şirket kayıtları oluşturulacaktır.

1.1.3. REQ3. Şirket kuruluşu için başvurma işlemi sırasında başvuru durumu "bilgi girişi" olarak atanacaktır.

1.1.4. REQ4. Şirket türü seçme işlemini Şirket Tescil Başvuru Sahibi yürütecektir.

1.1.5. REQ5. Bu işlem sırasında eTuzel Şirket Kurma Başvuru modülünde şirket türü kaydı oluşturulacak ve değiştirilecektir.

1.1.6. REQ6. Şirket türü seçme işlemi sırasında şirket türü, mevzuattaki tüm şirket türlerine göre oluşturulmuş listeden seçilerek atanacaktır.

1.1.7. REQ7. Serbest liman şirketi olma durumunu belirleme işlemini Şirket Tescil Başvuru Sahibi yürütecektir.

1.1.8. REQ8. Bu işlem sırasında eTuzel Şirket Kurma Başvuru modülünde serbest liman şirketi olma bilgisi kaydı oluşturulacak ve değiştirilecektir.

Figure 49 Excerpt from the requirements document of e-LegalEntity

The projects' technical contract differentiated from the generated document in the following ways:

- Some additional requirements were added for explaining the general properties of the system.
- Generated user requirements document was reorganized. In the generated requirements documents, requirement sentences were grouped by processes. That means, the headings and sub-headings for the requirements were identified by the processes and the process hierarchy. In this project, the customer required the requirements to be grouped by application systems (modules of the e-LegalEntity and e-Trademark systems). So, in the project's User Requirements Document, the generated requirements were reorganized under the headings for the modules of the system and the related requirements were copied under

relevant headings. This reorganization could also be conducted on the tool with some code development.

- Some of the Type 1 and Type 2 requirements sentences (as explained below) were united into one sentence to enhance readability, although the contracting organization analysts accepted that separate requirements would be better for testability.

As explained in UPROM methodology, there are three types of requirements sentences:

Type 1: In this type of sentence, the organizational elements conducting the related function are stated. In this case study, only the "carry out" action was used. This sentence type was generated to indicate the organizational elements who can carry out the functions.

Type 2: This sentence type is generated to express the CRUDL operations conducted on the entities and systems. During the modeling of FADs, the operations were used to express requirements and be a basis for size measurement, in conformance to the rules explained in UPROM methodology.

Type 3: The sentence is generated from the sentence-like statement already defined in constraint symbols on FADs. Looking it this way, it can be thought that generating these requirements from the models is similar to writing them manually to a requirements document, thus modeling does not add value. However, we observed that we were able to reveal these requirements by means of the guidance provided by the models. As requirements analysis activity was conducted on each FAD, one concentrated on what the system needs to achieve to successfully automate and complete the related function. As one thought about the function, entities and operations on those entities, it was easy to find out the constraints on changes of values or other business rules and needs of the system. Moreover, it was also easier to maintain those requirements as they are always found attached to the system, function and entities it belonged to. The interview results with the analysts of the project also supported this finding.

We made some updates to the requirement generation procedures of UPROM, as it was required by these projects. These changes were specific to e-LegalEntity and e-Trademark projects, and did not require any updates in the methodology. The list of changes is provided below. All of the required changes were reflected to algorithms of UPROM tool's requirements generation functionality.

- For the second type sentences, the statement "system" was replaced with the statement "module". This was because the application systems in this case study were used in "module" level; rather than just indicating that a function is conducted on the "e-LegalEntity" and "e-Trademark" systems, the modules of these systems were represented on the diagrams.
- For the first type and third type sentences, the part of the sentence that indicates the name of the application on which the function will be conducted was removed. The reason for this was the reorganization of the requirements

document, as explained in the above paragraph. As the requirements were already organized by module, the analysts indicated that there is no longer a need to provide the name of the module, as the related requirements were already grouped under that heading. The rest of the structure was not changed for the second type sentence.

- If the name of the application included "Web Service", then the statement for "module" was omitted, as we understood that this is an external system rather than a module of the system.

We need to emphasize that the updates in the generation of requirements for the project shall not be considered as an opposing application for the methodology, but rather it shows that the generation of the requirements is flexible and can be adapted to different needs of the systems and users.

Other than project specific updates, there was one need in the project that resulted in an update in the methodology. As explained in methodology section 3.3.1, if more than one organizational element is connected to a function on an FAD, we understand that those two organizational elements conduct (or whatever the connection type expresses) the activity together. However, we observed occurrences in this project where we want to state that one or more of the organizational elements can conduct the activity on her own. That means, any one of the organizational elements have the right to conduct the activity. To express this situation, a label of "1" was added to the connection name, as explained in the methodology, and the requirements were generated to state the one "or" the other organizational element can conduct the activity.

To ensure that the tool generated the requirements statements and the document as specified by the methodology, we provided an expert who is not involved in the projects' studies with the process model diagrams, UPROM requirements generation procedures and the rules specific to e-LegalEntity and e-Trademark projects. The expert analyzed the resulting requirements statements and confirmed that the requirements statements were generated conforming to the rules. The only problem he identified with the document structure is that, the order of the statements for a certain diagram is sometimes scrambled, which makes the document hard to understand. This issue is also identified in the observations explained below.

Data Collection – Metrics

The metrics for the generated requirements for e-LegalEntity and e-Trademark are shown in Table 13.

Table 13 Metrics for e-LegalEntity and e-Trademark requirements

Metric	e-LegalEntity	e-Trademark
Total number of generated functional requirements	363	177
Total number of requirements generated from constraints (constraints on FAD diagrams)	176	98

Number of requirements added manually	35	19
Total number of functional requirements	398	196
% of generated requirements	91%	90%
% of requirements generated from constraints	48%	55%
Number of modules	6	5
Number of requirements per module	66	39
Number of requirements per EPC diagram	26,5	32,7

The requirements added manually mostly include requirements on general structure of the system, interfaces with external systems and reference to the process models. 91% for e-LegalEntity system, and 90% for e-Trademark system of the requirements were generated, which is a high percentage and shows us that we are able to generate most of the functional requirements by means of the business process model and analysis diagrams.

Data Collection – Summary of Observations

The observations from the requirements analysis activities are listed below:

- Evaluation of how well the generated requirement statements were able to express the requirements
- Evaluation of the requirements for being complete, consistent, maintainable and traceable

These two items are evaluated together. As our group was conducting the analysis and modeling, we observed that EPC processes of the to-be system were very important to express how the system would work. By means of the process models, we had already established the infrastructure of the system. "Conformance to the processes" was also indicated as a requirement statement. The requirements generated based on these processes provided the requirements to be "complete" and "consistent".

In UPROM, the analysis conducted by FAD and ERD was guided by business processes. If we are sure that we have modeled our process in a complete and correct way, we have high chances that we evaluate every functionality of our system and analyze without missing any points.

This guidance helped a lot even for the "more manual" type of information embedded in FADs. The constraints added to FADs are different than the rest of the information added on FAD in a more mechanical way, like entities and the CRUDL like actions conducted on entities. They explain the rules, constraints, state changes that needs to be considered while that function is implemented, and the statements are written without any specific structure. As we focused on analysis on FADs, we went over these kinds of constraint type requirements and had a high chance not to miss them. Indeed, our customer was content with the resulting requirements for their well-structuredness and completeness. We observe that about half of the generated requirements originated from the constraints for both of the projects. This shows that determining such requirements is crucial to identify a complete set of requirements.

However, we also need to keep in mind that Type 2 requirements sentences is always counted as one, no matter how many operations are conducted within that function. Each operation carries an important knowledge of the system and shall be tested. But all operations are collected in one sentence for the sake of integrity and readability. If we were to count each of these operations as single requirement, the percentage of the requirements generated from constraints would be considerably lower.

The expert who reviewed and validated the requirements generated by the tool also had an observation on this aspect. For type 2 sentences, he criticized that it is not "fair" to regard all of the cases as just one requirement, regardless of the complexity of the FAD. He suggested that the sentence could be broken down into more than one sentence in some conditions, for example when there is more than one application. As we explained above, having all the operations in one sentence provided good readability, as the expert also agreed.

Another argument for type 2 requirements sentence by the expert was on the "during this operation" statement at the beginning of the sentence. He claims that for the sake of testability, this sentence should also start with the name of the function. We removed the function name to enhance readability, as repetitive function names following each other make focusing hard. However, if the developers plan to distribute different requirements to different developers and testers, the repetition can be preferred. It could be easily added by the tool if required. Similarly, the function names can also be removed for type 3 sentences if the stakeholders prefer so for readability.

Examining what operations need to be conducted during each activity revealed some functionalities in addition to the ones related to the basic function. For example, during the activity of "define company's communication information", the first requirement related to this activity was: company's communication information shall be created during this activity". Focusing more on the capabilities, we also think that if any communication information was created during this process, it can also be updated. This addition of operation also affected the size measurement, as will be discussed below.

As a result, having a set of processes that are approved by the contracting organization analysts and domain experts, we think that analysis activities have been well guided and we were able to achieve a set of requirements which are complete and consistent. For the maintainability and traceability, if we change the flow of processes, the functions, the entities, name of the entities or applications; it was practical to make changes in the models and regenerate the requirements. If the requirements were only written on a document, it would be hard to find out all effects of that change and reflect them on the document. Even a change in the name would require careful "find and replace" operations.

- Evaluation of generated requirements versus manually added requirements

We were able to generate most of the functional requirements, which are over 90% of total functional requirements. The rest of the requirements added manually were for explaining the high level structure and explanation of the system and interfaces. We observe that we were able to develop a very high percentage of the functional requirements by generation. For all the systems, we need to emphasize that additional functional requirements shall be identified which are not applicable to a specific process and function of the system, but rather focused on general functionalities of the system.

- Additional needs emerging while developing requirements

We observed the following points while analyzing and developing requirements, which can be used as input to future improvements of the methodology.

During the generation of the requirements by the tool, the order of the requirements for a model were sometimes scrambled due to the fact that the tool generates the requirements in order of placement on a process model. This reduced understandability of the requirements, but the problem was basically about the software tool rather than the methodology itself. The tool can be improved to list the requirements in the vertical placement order of the functions.

For all of the type 2 and type 3 sentences, we assume that the related operations (either the operations stated by CRUDL or as identified in the constraint) were conducted "during" the function, as stated by the word "sırasında" in Turkish. For a limited number of requirements we encountered in these projects, this was not the case. These operations are conducted when the function's execution is completed. The notation can be enhanced to depict "after execution" kinds of operations, and the requirements can be generated accordingly. The point here is that, considering that we conduct user level requirements generation, the complexity introduced to the notation and achieved enhancement must be evaluated and decided accordingly.

The customer united some (those that are short) of the Type 1 and Type 2 sentences to enhance readability. They accepted that leaving them separate are appropriate for "high quality" requirements, but preferred to keep united. This is accepted as customer specific preference for the project and is not generalized.

Data Collection – Summary of Interviews

The answers of interviewees for the questions regarding to requirements analysis activities and generated requirements statements are summarized below.

- All respondents agreed that the defined requirements were adequate to define system needs regarding the business processes, adding that this is applicable upon their existing knowledge. Moreover, they added that if we had conducted requirements analysis in traditional ways, it would be harder to achieve the same maturity level for the requirements and a more complete set of requirements.
  One of the analysts from our group indicated that he was able to indicate all functional requirements regarding to models, but we had to add additional requirements for general system properties (for which metrics are provided in

Table 13). He also mentioned that these were the parts where we spent a lot of maintenance effort, and if we could also associate this information to our models, it could be more maintainable. For future work, we can study on a feature to identify and model generic aspects of business application systems and develop models for them as part of the methodology.

- One of the respondents from the customer analyst team emphasized that even if it is not for achieving more complete requirements set, he would prefer this method because it is easy to disseminate requirements analysis tasks among the team of analysts. It made it also be possible to achieve the same style and maturity for requirements, as the method clearly guided the analysis activities and does not leave room for deficiencies caused by individual abilities.
- As a continuation of the previous answer, the interviewees also agreed that conducting requirements analysis focused on business process models helped to effectively reveal requirements.
- Considering the consistency of requirements throughout the document, both the domain experts and analysts indicated that as the requirements were clearly focused on well-defined aspects of the system identified by the models, the method did not allow inconsistencies and repetitions. If they were to conduct the analysis in traditional ways, as they wouldn't be able to structure the requirements this well, it would be a high possibility that there were inconsistencies and repetitions. We asked the contracting organization's analysts for their review results, they responded that those were not caused by significant inconsistency problems.
- The interviewees mentioned that the requirements document was smooth to read and understandable. Some of them indicated that at the beginning, they felt it hard to track, then they started following the document together with process models and were able to understand better. One of the contracting organization's analyst, who did not know at the beginning that the statements were generated automatically, stated that he could understand that there is a fixed pattern of sentences for each activity, but this did not create problem for readability but rather made sure that the required information is provided for each activity.
- The analysts asked to evaluate the time to conduct the analysis in this way or in regular ways, indicated that it would probably take more or less the same amount of time. However, when they were asked to evaluate maintenance efforts in case of changes, they strictly told that it would require far less effort to find out the effects of changes and update them with this method.

Generating COSMIC Functional Size Estimation

As mentioned in Activity 3 of the case study plan, the size measurement report was generated for both e-LegalEntity and e-Trademark projects by means of UPROM tool size measurement generation function, which generated the size measurement results automatically from business process models as explained in the methodology. The generated "Size Measurement Report" can be seen for e-LegalEntity in Appendix U, and for e-Trademark in Appendix BB of the technical report [207]. An excerpt from the generated report can be seen in Figure 50.

1.1.2. MSR2. SirketTuruSec

eTuzel Şirket Kurma Başvuru:

Entry Function Point: 2
Read Function Point: 1
Write Function Point: 2
Exit Function Point: 1
Toplam: 6 FP

1.1.3. MSR3. SerbestLimanSirketiOlmaDurumunuBelirle

eTuzel Şirket Kurma Başvuru:

Entry Function Point: 2
Read Function Point: 1
Write Function Point: 2
Exit Function Point: 1
Toplam: 6 FP

1.1.4. MSR4. SirketIsmiTanimla

eTuzel Şirket Kurma Başvuru:

Entry Function Point: 5
Read Function Point: 2
Write Function Point: 4
Exit Function Point: 3
Toplam: 14 FP

Figure 50 Excerpt from the FSE report of e-LegalEntity

Data Collection – Summary of Observations:

The observations related to functional size estimation are listed below:

- During modeling of FADs, we focused on adding related entities of a main entity that are defined on ERD. For example, when the Company entity is viewed, related entities of director, shareholder, secretary and person in authority are also viewed. This both brought clarity for requirements and provided better size estimation, as it is required to conduct separate operations on related entities according to COSMIC and UPROM results. This also conformed to the guidelines provided in section 3.2.4.
- There were different applications represented in FADs for both e-LegalEntity and e-Trademark systems. Some of these applications were modules of the system. Some of them were external applications or web services from which the systems obtain or send information. As these showed query and viewing of information from external systems, these data movements were indeed part of the systems. So, in size measurement, the data movements of operations on external applications were added to the size of the related module which exists on the related FAD, as explained as a rule UPROM FSE rules in section 3.3.2.
- A similar rationale applied when a module of the e-LegalEntity or e-Trademark system requests some information from an external system (modeled as a create operation for the request on the external system), and obtains an answer for the request (modeled as a view operation on the external system). Together with these operations, another operation was modeled that indicated that an entity is

created on the module for the answer. All of the operations were added as data movements for the related module.

- When a document is uploaded to the system, it was assumed that an entity is created in the system for that document. FADs were modeled in this way for the functions under the process “Add updated documents for data update”.
- Same FADs were used in multiple business processes. If exactly the same FAD was referenced and only one functional user existed on that diagram, the functional size was counted just once for these diagrams. An example for this is about providing the company approvals by the official. These three functions were utilized twice, when a new company is founded and when the information related to an existing company is updated. As these were conducted on the same module by the same organizational element, this was counted once. For the rest of the cases with more than one functional user, functional size was increased by the multiple of functional user number.
- In some activities, extra operations were added as explained in previous section. For example, in define company’s communication information activity, change as well as create operation was conducted on the communication information. This was caused by the realization that in addition to creating the data, the user also needs to change the value stored.
- ERD played an important role to determine how the functional size is counted. Although we emphasize that this is a conceptual level diagram, we focus on determining the “related entities”, as described in UPROM size estimation rules in section 3.3.2. By using the related entities, some unnecessary entries are avoided in the measurement. For example, for the function “List My Applications” the number of entries were calculated as 1, which would be 5 if we did not introduce the rule to eliminate extra Entry data movements for the related entities. For the function “Examine the information of the selected company”, there would even be 7 unnecessary entry data movements. Consider that, if the main entity “company” is added to the FAD, other entry movements caused by other operations were removed because company is related to other entities. However, if separate operations are conducted on each entity and company entity is not required for those, entry data movements are not eliminated. An example of this is for the process “find company according to search criteria”, where the entry movements are counted as 7.

Data Collection - Metrics

The estimated size for each module of e-LegalEntity and e-Trademark is provided in Table 14 and Table 15 respectively. Considering the size of the applications in ISBSG data, we see that many business applications lie in this size range. Total number of different data movements are provided in Table 16. Considering the distribution of data movement types according to domains [209], [210], the result of UPROM FSE was in harmony with the ISBSG findings for business application domain; having less number of write data movements and the rest being close to each other. Only the number of entries were higher than the averages. Considering this, we can claim that UPROM size estimation produced reasonable data movement weights for a business application.

Table 14 UPROM estimated size for software modules in e-LegalEntity

Module	FP Size
e-Tuzel Şirket Kurma Başvuru	297
e-Tuzel Şirket Başvuru Onay	370
e-Tuzel Şirket Bilgi Değiştirme Başvuru	265
e-Tuzel Şirket Listeleme	114
e-Tuzel Şirket Kapatma	56
TOTAL	**1102**

Table 15 UPROM estimated size for software modules in e-LegalEntity

Module	FP Size
e-Marka Marka Tescil Başvuru	119
e-Marka Marka Başvuru Onay	164
e-Marka Marka Bilgi Değiştirme	69
e-Marka Marka Değişiklik Onay	77
e-Marka Marka Tescil Yenileme	8
TOTAL	**437**

Table 16 Distribution of data movement types for e-LegalEntity and e-Trademark

Entry	Read	Write	Exit
e-LegalEntity			
337	293	202	298
31%	27%	18%	27%
e-Trademark			
119	112	89	117
27%	26%	20%	27%

Data Collection – Summary of Interviews:

The questions for the evaluation of size estimation were only asked to the analysts. The customer analysts did not previously know about size measurement and COSMIC standard. Thus, we provided a brief explanation of the standard and idea before starting the interview. Although they were not able to technically evaluate the results of size estimation, they explained that they would like to use such an information while contracting the development work. Indeed, effort estimation based on the estimated size was also prepared for the two projects and used for bidding. Furthermore, contracting company wanted to extend size estimation to other e-government projects so that they can have a repository of planned and actual size and effort values for the future.

Validation for e-LegalEntity Size Estimation:

Case study set 1 was already conducted to ensure that we can arrive at "reasonably accurate" results by UPROM FSE method. We had discussed there that those cases are defined with software level requirements. However, we don't expect this much information to be available at the system analysis level conducted with UPROM. Thus, with the user level requirements identified in these projects, we conducted a partial comparison to evaluate the estimated results by UPROM with the measurement results by an external expert. Still, it needs to be kept in mind that as the requirements were not in software level, external expert also needed to make assumptions. So, her COSMIC measurement results can also be regarded as "estimation" and is not expected to be as accurate as COSMIC based size measurement from software requirements.

The functions on "SirketKurma.epc" diagram are provided as an example to compare expert results and UPROM estimation results as seen in Table 17.

Table 17 Comparison of manual size measurement with UPROM FSE results

Function Name		Data Movement Type				Total	Absolute Difference
SirketKurulusIcinBasvur		E	R	W	X		
	Expert	1	0	2	2	5	1
	UPROM	1	0	2	1	4	
SerbestLimanSirketiOlmaDurumunuBelirle	Expert	2	1	2	2	7	1
	UPROM	2	1	2	1	6	
KurulusBasvuruHarciOnayla	Expert	1	1	2	1	5	2
	UPROM	1	2	2	2	7	
SirketTuruSec	Expert	1	1	1	2	5	1
	UPROM	2	1	2	1	6	
SirketIsmiTanimla (w/o web service)	Expert	2	1	2	3	8	2
	UPROM	4	1	3	2	10	
Total	Expert	2	1	2	3	31	2
	UPROM	4	1	3	2	33	
	Deviation in absolute error for each functional process						27%
	Deviation in total size						6%

For the whole sample, the deviation of the absolute error of each functional process is 27%, and deviation in total size is 6%. Considering the reference limits identified in case study 1, even the deviation in absolute error is below the identified limit (30%) and we conclude the estimation to be reasonably accurate.

Generating Process Documentation

Within the scope of UPROM, process documentation included three artifacts: Process definition document, business glossary and process improvement list. Generation of process KPI list was out of scope for these projects.

Terminology definitions, process meta-data, descriptions for the functions and improvements were added where applicable during modeling phases. We rescanned the models to find out if any more definition, description and improvements need to be attached to the processes.

3. Şirket Kuruluş Belgelerini Yükle

/01-Sirket/01-SirketKurma/01-SirketKurulusBelgeleriniYukle/SirketKurulusBelgeleriniYukle.epc/

1 Süreç Bilgileri

Sürecin amacı: Şirket kurma işlemleri sırasında yüklenecek belgeler ile ilgili işlemleri tanımlama
Sürecin kapsamı: RKMMD Şirketler Mukayyitliği ve Serbest Liman ve Bölge Müdürlüğü kapsamında kaydedilen tüm şirketler
Sürecin durumu: Müşteri tarafından onaylandı
Versiyon: 1.0

2 Süreç Sorumluları

		Sorumluluk Tipi				
Sorumlu	Tipi	R	A	S	C	I
Şirket Tescil Başvuru Sahibi	Dış kişi	X				
Şirket Kurma Başvuru Sahibi	Dış kişi	X				

3 Girdiler

İsim	Tipi	Kaynak	Link
tescil beyanı	Doküman	Kaza Mahkemesi	/\01-Sirket\Dokumanlar\Ek4-16.ms1.pdf
karakter belgesi	Doküman	Polis Müdürlüğü	

5 Başlama Koşulları

Olay	Bitiş Olayı Olduğu Süreçler	Adresleri
Şirket kuruluş / bilgi değiştirme bilgilerinde eksik yok	SirketBasvurulariOnay.epc	/01-Sirket/03-SirketBasvurulariOnay/
	BasvuruBilgileriniKontrolEt.epc	/01-Sirket/03-SirketBasvurulariOnay/01-BasvuruBilgileriniKontrolEt/

6 Bitiş Koşulları

Olay	Başlangıç Olayı Olduğu Süreçler	Adresleri
Başvuru durumu "bilgi girişi onaylandı, mukayyitten onay bekleniyor" durumunda	BasvuruBilgileriniKontrolEt.epc	/01-Sirket/03-SirketBasvurulariOnay/01-BasvuruBilgileriniKontrolEt/

7 Aktiviteler

7.1 Tescil beyanı gir

Sorumlular:

Şirket Tescil Başvuru Sahibi - yürütür

Girdiler:

tescil beyanı

Bilgi Sistemi:

eTuzel Şirket Kurma Başvuru

Kaza Mahkemesi'nden onaylatılan Tescil Beyanı, sisteme girilir.

Figure 51 Excerpt from e-LegalEntity process definition document

2 Sistem Tanımları		
İsim	Kısaltma	Tanım
eTuzel Şirket Listeleme		Mukayyitin şirketleri listeleyebildiği, çeşitli kriterlere göre arama yapabildiği ve detaylı bilgileri görüntüleyebildiği; şirketle ilgili kişilerin bağlı olduğu şirket bilgilerini görüntüleyebildiği yazılım
eTuzel Şirket Bilgi Değiştirme Başvuru		Şirket yetkili kişisinin kendi şirketiyle ilgili bilgileri değiştirmek, bu bilgilerle ilgili güncel belgeleri girmek, başvurusunun takibini yapmak, durum değişikliklerini ve mukayyitten gelen bilgileri görüntülemek için kullandığı yazılım

3 Genel Tanımlar		
İsim	Kısaltma	Tanım
RKMMD	RKMMD	Resmi Kabz Memurluğu ve Mukayyitlik Dairesi
Şirket		Şirketler (Sınırlı Sorumlu)Yasası veya 1949 Şirketler (Güvence ile Sınırlı)Yasası uyarınca kurulmuş ve tescil edilmiş bir şirket.
Şirket vekil bilgileri		Vekalet verilen gerçek kişi için: Ad, soyad, kimlik no, giriş tarihi, çıkış tarihi, ikamet adresi, telefon, e-posta
Ikamet		
Gerçek kişi kimlik bilgileri		

Figure 52 Excerpt from e-LegalEntity business glossary

İsim	Açıklama	Fonksiyon	Diagram
Karakter belgesi durumu	Polis Müdürlüğü ile iletişim sağlanabilirse vatandaşın gidip karakter belgesi alması, yüklemesi gibi ihtiyaçlar ortadan tamamen kalkacaktır.	Karakter belgelerini yükle	01-Sirket/01-SirketKurma/01-SirketKurulusBelgeleriniYukle/SirketKurulusBelgeleriniYukle.epc
Vergi bigileri girişi	Vatandaş ya da tüzel kişinin vergi bilgileri durumu GVD'den otomatik alınacağı için kendisinin belge getirmesi gerekmeyecektir.	Vergi Güvenirlilik Belgesi gir	
Dış onayları alma	Dış onaylar, web servis bağlantısı olmasa da BelgeNet üstünden tamamlanacaktır. Böylece onay alma işlemleri hızlandırılabilecektir.		01-Sirket/01-SirketKurma/02-SirketKurulusDisOnaylariniAl/SirketKurulusDisOnaylariniAl.epc

Figure 53 Excerpt from e-LegalEntity process improvement list

Process documents were generated for e-LegalEntity and e-Trademark systems by means of UPROM tool process document generation functions. First of these functions generated process definition document covering each process in the project, and formed a collection of process definition documents, as explained in section 3.3.4. The second function generated a single business glossary covering the definitions for the whole project as explained in section 3.3.5. An entry in the business glossary may exist as an object in all processes of the modeling project. Hence, reader of any process may consult to the same business glossary and understand what that means as the designers of the system and other readers. By using the second function, a list of improvements as assigned on the models were prepared. By using this report, the analysts and modelers could reveal the points where improvements were identified in all of the project in a single list, obtain metrics and evaluate the benefits of the system to be automated.

The generated “process definition document” can be seen for e-LegalEntity in Appendix V, and for e-Trademark in Appendix CC of the technical report [207]. The business glossary can be seen for e-LegalEntity in Appendix W, and for e-Trademark in Appendix DD of the technical report [207]. Process improvement list can be seen for e-LegalEntity in Appendix X, and for e-Trademark in Appendix EE of the technical report [207]. Excerpts for these documents are provided in Figure 51, Figure 52 and Figure 53 respectively.

To assure that the tool generated the documents as defined in UPROM, two researchers were commissioned to check the generated documents. We provided the researchers, who were not involved in project activities, process models and UPROM tool (so that she can observe descriptions attached to the process models), the description of the generation procedure and the generation principles specific to these projects as explained below. One of the researchers checked process definition documents and the other checked business glossary and process improvement list. The researchers approved that the generated documents conform to the expected outputs when procedures are applied. The projects’ process definition document and business glossary documents deviated from the method identified in methodology section in the following ways.

- All of the organizational elements in both of the projects were connected to functions (either on EPCs or FADs) with “carries out” type of connection. The other four types of connections between an organizational element and function was never used. The second part of the process definition document, “Process Responsibilities” heading was simplified, the type of connection was removed from the table.
- The 11th section of the process definition document, “KPI List”, was not generated because there were no KPIs defined in the processes. Definition of KPIs were not in the scope of these projects.
- There was no need to use the “technical term” symbol throughout the projects, as all necessary definitions were already provided by means of document or entity symbols. So, no business glossary entry was generated from the technical term symbol. This didn’t affect the business glossary, as the required definitions are already captured by other symbols.

Data Collection – Metrics

The metrics on the process definition documents, business glossary and process improvement list outputs are provided in Table 18.

Table 18 Metrics on process documents for e-LegalEntity and e-Trademark

Metric	e-LegalEntity	e-Trademark
% of process models for which general process descriptions (process metadata) are added	100%	100%
% of functions for which descriptions are added (except the ones referencing sub-processes)	96,5%	95%
Number of entries in business glossary (includes the ones which don't have technical term description but added because it is a part of a high level entity)	82	40
Ratio of business glossary entries per total number of entities in the project	89% (82/92)	74% (40/54)
Number of business glossary entries per EPC diagram	5,5	6,7
Number of improvements	18	7
% of improvements per application systems in the project	3	1,2
Number of improvements per EPC diagram	1,2	1,2

Observing the metrics, general process descriptions were added as process meta-data for both of the projects. Also, many of the functions were detailed with additional descriptions. We can see that project stakeholders used the functionality and found this helpful to further describe the processes.

Considering the business glossary entries, the ratio of number of items in business glossary is high when compared to all entities in the project. This means that a definition of a technical term was required by the project team for most of the entities. However, one must keep in mind that most of the entries did not have descriptions but added to business glossary because it was a part of the high level entity. Still, addition of these items to the dictionary was important to provide information on the data.

We see that only a few number of improvements were identified in the system. Considering that these improvements were planned to be achieved by means of the applications to be developed in the project, the ratio of improvements per application system is a reasonable metric and the value is low. The reason for this is not because there were only this much amount of improvement provided by the system. Process models in both of these projects were designed for the to-be systems, thus already defining what is to be achieved. Also, there was no request by the management to report a list of requirements. So, the project team did not need to enter detailed information on the opportunities during analysis and modeling activities.

Data Collection – Summary of Observations:

- Evaluation of how well process definition document meets requirements of process documents developed traditionally

The information covered in process definition document generated in this study was gathered under eleven headings for each of the processes. The most important part of this document was the “Activities” heading which we observe that took the most space in process documents. Some of the process documents was even composed of just this part.

In summary, we observe that generated process definition document mostly fulfills the content of a manually developed process definition document, whereas provides some additional information which is required to understand the process better but usually avoided because the related information is hard to maintain manually and dependent on many other processes. In these projects, we see that project definition documents provided description in a format that is comprehensible by different types of stakeholders, provided additional descriptions not observable on models, and presented in tabular format which makes it possible to grasp the process’s relation with other processes at one glance.

- Evaluation of the usefulness of the business glossary to provide a common understanding among stakeholders

As discussed above, we observe that in most of the process definition documents, definitions and abbreviations are placed as part of the document. However, the problem usually is no one can reach a full set of these definitions and see if they are compatible with each other. In e-LegalEntity and e-Trademark projects, the list of definitions was important for different stakeholders. The users of the systems are both citizens, representatives from legal entities and officers from different organizations. It was important to provide a central definition for all of them and make them understand the same thing from the concepts. Also, the aggregation information covered in ERDs, which will not be read by the users of the system, are presented in the business glossary. This also assured that for everyone to understand what is covered within an entity.

The external expert who reviewed the tool outputs for business glossary provided a suggestion to include all of the organizational elements and applications, maybe all entities and information carriers, even if they don’t have any terminology description added. She argues that those objects, even if the terminology isn’t defined for them yet, has an important role as data in the system and must be observed in the business glossary list. We believe that this is a valuable suggestion and can be easily added as a functionality to the tool if required by users.

- Evaluation of benefits for improvement list reporting

In many projects, to make the results of the improvements enabled by process analysis visible to the stakeholders, especially the management, personnel require to note down the possible improvements introduced by the analysis and get a listed report of those. In e-LegalEntity and e-Trademark projects, all of the system was redefined in to-be processes, and technical contract documents were developed to describe the requirements of the systems based on the process models. Because of this, the team did not require to obtain a detailed improvement list, as all of the improvements were already visible to the stakeholders. However, a short report is generated to highlight the important points and delivered as an output of the project.

- Evaluation of maintainability and traceability

As discussed in the above bullets, there are many information which experts avoid to add in process documentations as it is hard to maintain. In e-LegalEntity and e-Trademark projects, processes were updated and related process documentation are re-delivered many times. As long as the objects were kept, reorganization of the flow or addition of new objects were no problem for the documentation. Also with the support of unique object property, information on every object was kept up to date with every update of the processes. All of the documents were generated easily with the tool functionality and no additional effort was required.

Data Collection – Summary of Interviews

During the interviews, all interviewees were asked to evaluate all three process documentation outputs together. Answers are summarized below.

- All of the interviewees indicated that process definition documents are critical for users of different backgrounds and field of studies. However, all interviewees also told that they would like to have both process models and process definition documents to understand and follow a process, both of them are critical and none of it has priority over the other. This shows that process definition documents are not only essential for less technical personnel, but more experienced and technical personnel too.
- The interviewees also added that the business glossary covered essential information together with the process definition documents. Analysts moreover appended that this will definitely be the starting point to develop a detailed data dictionary for software development or other purposes.
- Interviewees indicated that the generated documents covered what they like to see on a process document, and moreover cross-process information on the document provided additional analysis opportunities which are hard to see on traditional documents and on the process models.
- The interviewees stated that if we were to develop these kinds of documents manually, it would take more time and we would have a high probability of missing information as we wouldn't be able to do a structured analysis. But the big difference would be on the maintenance effort, they added. For the analysts, it was almost impossible to be sure that they reflected the effects of a single change on multiple points of the process definition. But with the models, they could find out, update and regenerate the documents to end up in complete and consistent documentation.
- All of the interviewees indicated that business glossary is critical to provide a common understanding between different stakeholders. They indicated that regularly, it is hard for them to assure that they end up with a complete set of

definitions, but they could find out all terminology and assign unique definitions by using the tool.

- The process improvement list was introduced to the interviewees as a pilot version after the project analysis is completed, because such an output was not required in the project. We asked them to evaluate the list as an idea but not as a complete list. However, the interviewees responded that it is a very good idea and we needed to record and report the benefits to be acquired from the project; thus this could be an essential part of the project outputs. They also stated that we would be able to add this information to the models during process modeling, and would prefer to have done so.
- As the last question, we asked the interviewees how they find the process of conducting single process analysis activity and generating all these outputs automatically using this information. Many interviewees indicated that it is impressive to achieve so many outputs all together, this method provides efficiency for the overall effort, assures consistency among products as they are sourced from the same information and can be regenerated all in case of changes in the source. Two analysts specifically added that as the method guides you well to conduct the analysis, it assures you to reveal necessary information thus develop complete outputs and achieve a consistent quality even if you employ more than one analyst for the work.

Evaluating the Overall Results

As a result of these case studies set, we answer the research questions in the following way:

CSQ1: We were able to follow UPROM as the methodology and utilized the related tool for e-LegalEntity and e-Trademark projects to analyze and model the processes, generate process documentation, develop user requirements and estimate the size of the software to be automated. All these artifacts were developed by using the information captured by business process models. We utilized the tool's functionality to automatically generate the artifacts from process models.

CSQ2:

- Both the observations and interview results support that the process models were able to achieve coverage for the information required in the project, which was hard to achieve if traditional approaches were followed. The models conformed to modeling guidelines, the project structure helped the analysts to analyze and organize the models.
- 90% of the functional user requirements in the technical contract were generated automatically. These requirements were assessed to be accurate and consistent by different stakeholders of the project. As it is the single source of information, all artifacts were traceable to process models. Maintainability was high as conformed by all stakeholders, considering the cases where updates were required for the artifacts.

- Early size estimation of the systems were achieved. The analysis of deviation with manually measured COSMIC results showed that the estimated size is reasonably accurate.
- Process definition document and business glossary, as generated by the tool, proved to be helpful for the stakeholders to understand the processes and achieve unique comprehension. Process improvement list, though prepared in later phases, was found to be a critical report to visualize the benefits of the study by the analysts.

4.4. Case Study 3 – Public Investment Project (PublicInvest) for Ministry of Development

In this section the third case study of the study is explained. First, the background information for the project is explained. Case study questions are defined. Then, the plan for the case study conduct is given. Sources of evidences are identified. In the last section, conduct of the case study, data collection and analysis results are provided.

4.4.1. Background

The objective of BİHAP program was implementation of research and development services for information map of the Ministry of Development, processing the information map items electronically after extraction, visualizing the information map items and relating them to the business processes, and developing a methodology and model on information map that can be applied by other public institutions. The program was conducted as a consortium of three organizations. One of the projects included analyzing and defining business processes for one of the main function areas of the ministry: Public Investments.

The public investment processes mainly covered development and publishing of investment programs, deciding on the projects to be executed within the programs and execution and completion of the investment programs. This part of the project was named as “KB_Kamu_Yatirimlari” (Public Investments - PublicInvest). Process analysts worked on these processes together with domain experts using UPROM tool and following UPROM methodology to analyze and model the processes. Project scope did not include automation of the related processes. The outputs required by the project were business process models, process definition documents related to those models, business glossary and a report for improvements. The whole project already included definition of data in detail and generation of an information map. But before that work was completed, they required to obtain a list of terminology utilized through the defined business processes, which was to be the basis for ontology studies in further steps. Business glossary was also generated by UPROM to meet this need. Considering the requirements, this project was selected as the object of this case study.

As researchers, the authors of this study were not involved in the process modeling and analysis activities of this project. Before start of the project, a training session were conducted to train the analysts on the usage of UPROM methodology and the tool, and necessary support was given when necessary. The resulting outputs; business process models, process definition document, process improvement list and business glossary; were collected upon finalization of the study and data was collected using those. Additionally, interviews were conducted with the analysts and domain experts from the organization to evaluate the results.

4.4.2. Objective

In order to answer the general research questions, this case study focused on integrating BPMod and process documentation practices. Thus, to contribute to answering the research questions (provided in 4.1.2), the research objective of this case study was to examine, for PublicInvest project, if UPROM can be used to analyze and model processes and generate process documentation including process definition documents, business glossary and process improvement list all together and based on business process models. Furthermore, it is aimed to evaluate if these artifacts better than the ones developed with traditional methods in terms of consistency, completeness, maintainability and traceability.

4.4.3. Case Study Plan

The objective of this case study set is to apply UPROM to develop the necessary artifacts required by the project, collect and analyze the data and answer the research questions. The planned list of activities to be conducted for this case study is the same with Case Study Set 2, apart from the steps 2, 3 and 4; as the project's scope does not cover automation of processes.

4.4.4. Sources of Evidences

The objects of this case study is the PublicInvest project. The subjects of this study from which data is collected are the analysts and an end user with which interviews are conducted. The sources of evidences to be collected during this case study is listed under "data to be collected" column of the case study plan table. The types of sources of evidences can be summarized as follows:

Documentation: Existing documentation to understand the as is processes, like laws, can be listed under this type of source of evidence.

Interviews: Interviews were conducted in this study to evaluate the methodology and results by both analysts and domain experts. A different point of this study is that, the researchers developing UPROM were not involved in the project. By means of the interviews, we are able to gather ideas of different types of stakeholders which are totally immersed in the project activities. The interview questions can be found in Appendix FF of the technical report [207].

Participant Observation: Participant observation was not conducted by researchers but obtained from the analysts of the project.

In this case study, we utilized metrics collection for quantitative analysis, and observations and interviews for qualitative analysis of the results.

4.4.5. Conduct of the Case Study, Data Collection and Analysis

Conducting Descriptive Business Process Analysis and Developing ERD

Business process analysis was conducted using available documentation of the organization and together with domain experts by means of workshops. FT, EPC, OC and ER diagrams were used to model the processes. Business process models were developed by three analysts which are not involved in UPROM research activities. The structure of the project is shown in Figure 54.

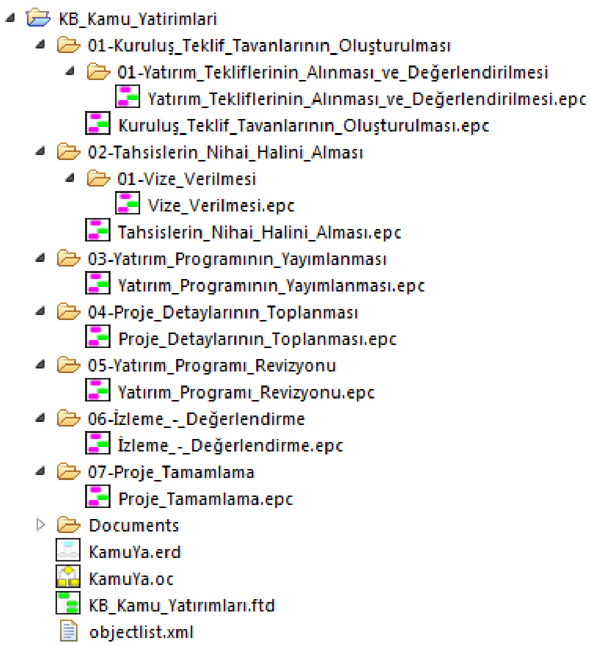


Figure 54 Modeling project structure of the PublicInvest

As seen in the figure, under the main project folder, there are three models:

- KamuYa.erd: Provides high level entities and their relations in the system (shown in Appendix HH of the technical report [207]).
- KB_Kamu_Yatırımları.ftd: The process map file for the project. All the process models in one level lower are linked to this process map.
- KamuYa.oc: Organizational chart that includes all organizational elements and their relations within the scope of the project.

The folders were organized in numbers to provide a reasonable ordering of the diagrams. The names of the EPC models in order and in hierarchy are as follows:

- Establish organizational proposal limits
    - Obtain and evaluate investment proposals

- Finalize allocations
    - Provide visas
- Publish investment program
- Collect project details
- Revise investment program
- Monitor and evaluate
- Complete project

In the modeling of the project, the following symbols were used:

- **EPC:** Event, Function, Process Interface, Application, Organizational Elements (Organizational Unit, Position), Document, Connectors (And, Or, XOR)
- **FTD:** Function
- **ERD**: Entity, Relationship, Generalization
- **OC**: Organizational Unit, Position

The structure of the project conformed to the procedures of the methodology. Project structure was validated by UPROM tool without any errors. Live validation functionality was utilized and the EPC models were developed free of semantic errors.

Data Collection – Metrics

The metrics on the process models are provided in Table 19. Detailed metrics for EPCs in PublicInvest project are provided in Table 20.

Table 19 Metrics for PublicInvest business process models

Metric	PublicInvest
The deepest level	2
Number of EPC diagrams	9
Number of FTD diagrams	1
Number of function symbols on EPC diagrams	58
Number of function nodes per EPC diagram	6,4
Total number of unique control flow nodes	135
Total number of unique control flow nodes without event nodes	89
Average number of total control flow nodes per EPC diagram (without events)	9,9
Average number of connector nodes per EPC diagram	3,1
Average number of arcs per EPC diagram	15

The scope of the project included analysis of the related business processes in a higher granularity and automation of the processes were not envisioned. In conjunction to this, we observed that the number of total control flow nodes on EPCs were not high, while the number of organizational elements and information carriers were relatively high. Still because of the same reason, multiple usage of start and end events were limited. Only 2 diagrams had multiple start and end events. Considering the criteria for the nodes provided in "7 Modeling Guidelines" [9], we observe that any process model was far less than the suggested total number of nodes, 48; 15 being the highest and 9,9 the average. Number of connectors for one of the processes was high, 8; but the rest was again small in number. Number of functions and number of arcs were also lower than the suggested values for all process models. We can say that these models complied with the 7PMG [9] for the size of the models (for G1, G2 and G7).

As suggested by [9] (G4), models shall be structured for splits and joins. Splits and joins in some of the processes were not paired, for example when different paths of a split is connected by a single join. Or routing element was used in 5 split-join pairs. We also interpret this as a result of higher level modeling, being harder to model exact alternatives. The depth of the hierarchy was 2, not going too deep into the details of the models. All of the activities were labeled in an imperative way, as suggested by the guideline (G6) and UPROM guidelines.

Table 20 Detailed metrics for EPCs in e-PublicInvest

	Function	Connector	Prc.Int. (instance)	Control flow nodes w/o events	Event	Control flow nodes	App.	Org.El.	Inf.Car.	Bus. Rule	Total	# of arcs	Start Event	End Event
Kuruluş_Teklif_Tavanlarının_Oluşturulması	14	2	1	17	7	24	2	9	14	0	49	24	1	1
Yatırım_Tekliflerinin_Alınması_ve_Değerlendirilmesi	10	5	0	15	9	24	1	2	22	0	49	22	1	3
Tahsislerin_Nihai_Halini_Alması	5	2	1	8	5	13	1	6	4	0	24	12	1	1
Vize_Verilmesi	6	8	0	14	6	20	1	4	4	0	29	21	1	1
Yatırım_Programının_Yayımlanması	4	0	0	4	3	7	1	2	5	0	15	6	1	1
Proje_Detaylarının_Toplanması	5	2	0	7	3	10	1	3	3	0	17	10	1	1
Yatırım_Programı_Revizyonu	9	5	0	14	7	21	1	5	4	0	31	26	1	1
İzleme_-_Değerlendirme	3	4	1	8	4	12	1	3	28	0	44	13	2	2
Proje_Tamamlama	2	0	0	2	2	4	0	2	2	0	8	3	1	1
Total	58	28	3	89	46	135	9	36	86	0	266	137		
Average	6,4	3,1	0,3	9,9	5,1	15,0	1,0	4,0	9,6	0,0	29,6	15		

Data Collection – Summary of Interviews

Interview results are provided for all topics together after the next section.

Generating Process Documentation

Within the scope of this case study, process documentation included three artifacts: process definition document, business glossary and process improvement list.

During business process modeling activities of PublicInvest processes, definitions of the terms were recorded on the models. Also, descriptions for the functions were added where required. The number of functions with description was limited, as process analysis was not conducted in low level of detail for the project. Improvements were also added on the diagrams during modeling activities. Business process diagrams were provided to the domain experts as a report and processes were discussed through the models in the workshops. Upon the finalization of the models, process definition document, business glossary and process improvement list documents were generated by UPROM tool.

Business process models and ER diagram can be seen in Appendix HH, process definition document can be seen in Appendix II, business glossary can be seen in Appendix JJ and process improvement list can be seen in Appendix KK of the technical report [207]. the documents are provided as generated from UPROM tool.

The project's process definition document, business glossary and process improvement list documents deviated from the artifact generation procedures in the following ways.

- All of the organizational elements in both projects were connected to functions (either on EPC or FAD) with "carries out" type of connection. The other four types of connections between an organizational element and function were never used. The second part of the process definition document, "Process Responsibilities" heading was simplified, the type of connection was removed from the table.
- The 11th part of the process definition document, "KPI List", was not generated because there were no KPIs defined in the processes. Definition of KPIs were not in the scope of these projects.
- Process Improvement table was decreased to three columns, "description" column was removed as the analysts did not use description attribute for improvement symbols on the models but rather provided all information as the name.

Data Collection – Metrics

The metrics on the process definition documents, business glossary and improvement opportunity list outputs are provided in Table 21.

Table 21 Metrics on process documents for PublicInvest

Metric	PublicInvest
% of process models for which general process descriptions (process metadata) are added	100%
% of functions for which descriptions are added (except the ones referencing sub-processes)	29%
Number of entries in business glossary (includes the ones which don't have technical term description but added because it is a part of a high level entity)	93
Ratio of business glossary entries per total number of entities in the project	100%
Number of business glossary entries per EPC diagram	10,3
Number of improvements	19
% of improvements per application systems in the project	5
Number of improvements per EPC diagram	2,1

We observe that although process metadata was added for all diagrams, less than third of the functions were enriched with descriptions. This was due to the nature of the project, analyzing on business processes on higher level and focusing more on data specific elements. Thus, we see that the number of business glossary entries per EPC diagram was far higher than that of e-LegalEntity and e-Trademark projects (5,5 and 6,7 respectively). The percentage of number of improvements was also higher in this case. Considering the metrics, we conclude that data centric analysis perspective of the project affected the structure of the models; with less number of functions and details and high number of information carriers, entities and improvements. This enabled us to observe that UPROM can be utilized for process analysis activities with different approach and aims.

Data Collection – Summary of Interviews

The interviewees for PublicInvest project were 3 analysts from the contracting organization and one domain expert from the ministry. During the interviews, all interviewees were asked to evaluate business process modeling activities, resulting business process models and process documentation outputs together. Analysis of interview questions by different types of project stakeholders are provided below. A general evaluation of the interview results and the overall findings are presented for each topic. Interview questions are provided in Appendix FF of the technical report [207]. Interviews were conducted in a semi-structured way and the conversations are recorded with a voice recording device. The summary and highlights of the interviews for each question are provided in Appendix GG of the technical report [207].

- All of the analysts were experienced in business process definition and usage of different BPMod notations.
- The analysts agreed that the usage of certain symbols and notation aspects on diagrams enabled them to analyze processes better. These are applications, same events in different processes, sub-diagrams and process interfaces. They indicated that they didn't use business rule, because they needed high number of business rules for a few functions, which would make the diagram look too crowded. They preferred to add those rules as descriptions to the functions.
- The analysts mentioned that usage of unique objects supported them for efficient analysis and provided ease of maintainability. They indicated that they especially needed to update the names of organizational elements and information carriers a lot of times, and the unique property provided easy update for them. Also, they added technical term definitions from one instance and maintained them at any place on the diagrams.
- For the hierarchical structure, the analysts explain that, at first they started modeling in a flat structure. Then they were "forced" to make a hierarchical structure by the tool. They indicated that it was easier for them to organize the sub-processes in pieces and analyze them in this way. One of the analysts indicated that he liked seeing all the diagrams at a glance when it was flat, but he preferred the organized way of hierarchy now, especially when there are a lot of diagrams.
- Although so many instances of multiple start and end events were not used on the diagrams, analysts think that it enhances their modeling capabilities.

- The analysts mentioned that the domain experts they work with were BPM-illiterate at the beginning of the project and it took them a lot of time to make them familiar with the notation, in addition to the training sessions. They indicated that still, business process models made their jobs far easier to analyze the processes together with domain experts and they wouldn't prefer natural language definition. Maintaining the processes would be very hard for them if they had defined the processes by natural language.
- UPROM tool forced the analysts to develop the models conforming to specific meta-model definitions. Two of the analysts indicated that at first, he had problems with connection limitations and wanted to use the symbols in other ways. But then, after learning the definition of the related symbol, the limitations were meaningful for him. So, the meta-model helped the analysts to understand the notation and use it in conformance with the methodology.
- The analysts indicated that business process models alone were enough for them to analyze and follow the processes, as the project's scope did not include analysis of the processes in very low detail. However, domain experts absolutely liked process definition documents and liked to read the processes using both the models and the documents together. The analysts emphasized that it would be very hard for them to analyze the processes and end up with complete and consistent set of definitions with natural language, and to keep such a documentation updated when the changes occur.
- The project was part of a bigger project which aimed to develop an information map for the operations of the ministry. In this part of the project, the analysts tried to utilize process models to identify the related informational aspects. Thus, attaching terminology definition to related object types and adding technical term symbol when additional terms were required helped them to identify all kinds of entities in the processes. They indicated that business glossary was helpful for them to obtain a full list of items in the processes and analyze if there are any missing. Domain experts also liked to use the business glossary artifact to understand the processes.
- Identifying the possible improvements to be achieved by analyzing and improving the processes was a critical task in the project. The analysts utilized improvement symbols to record such data from the beginning. They indicated that it was an easy way of fulfilling this task for them. Domain expert was also content with the resulting report.

Evaluating the Overall Results

As a result of this case study, we answer the research questions in the following way:

CSQ1: UPROM and the related tool were utilized successfully in PublicInvest project to analyze business processes and generate process documentation including process definition document, business glossary and improvement opportunity list. The analysts of the project confirmed that they were able to develop all artifacts based on the information captured on the single source of business process models. All generated artifacts were utilized as project deliverables. We observe that in a project with a different nature, where the automation is not the aim and processes are analyzed in a higher level, UPROM was used successfully.

CSQ2:

- Both the analysts, and domain experts accepting the artifacts as deliverables of the project agreed that process models and the artifacts covered the information required. The analysts confirmed that the project structure supported them to analyze the processes. The process models conformed to modeling guidelines.
- Analysts and domain experts confirmed that generated process definition document and business glossary were helpful to understand process models and achieve a consistent understanding of the process concepts and the terminology. Major consistency and completeness problems were not identified by domain experts in their reviews. Process improvement list was used by analysts as a report to demonstrate the benefits of the project. Analysts indicated that maintainability of the artifacts was high.

4.5. Case Study 4 – Integrated Information System Project for METU Campus (METU-IIS)

In this section the fourth case study of the study is explained. First, the background information for the project is explained. Case study questions are defined. Then, the plan for the case study conduct is given. Sources of evidences are identified. In the last section, conduct of the case study, data collection and analysis results are provided.

4.5.1. Background

METU has 24.000 students currently enrolled. There are 40 undergraduate departments and 160 graduate programs. METU also has 21 interdisciplinary research centers. More than 5.000 personnel are working for the university. A large number of IT systems have been developed since the establishment of the university Computer Center. However most of these IT systems run independently, not communicating with each other and using various technologies for data storage and communication. As a result of this crowded, complex environment, problems emerged such as out-of-control duplication of data, non-standard communication, lack of control over IT service levels and very high maintenance costs.

The Integrated Information System (IIS) project was initiated in 2009 by Computer Center in order to solve these problems. IIS aims at integrating the existing IS applications in accordance with the university strategic plan. Initially Computer Center developed a business process map consisting of all the business processes of METU. These processes were prioritized in line with the master project plan.

Since the beginning of the project, the activities of analyzing and modeling of business processes and eventually developing software running on automated business process models are conducted iteratively for each process module. More than 300 person-months are utilized in the last 3 years.

The method for defining business processes can be summarized as follows. First all stakeholders involved in the business process are contacted. Then modeling experts start analysis sessions with the stakeholders. The analysis team puts in additional effort to produce process definitions documentation. Then the BPMN models of the processes are developed in accordance with the textual definitions. The data elements are fed into the university business glossary. In compliance with the data element definitions, the web services are implemented and SOA mediation layer are integrated with them. Then the software models associated with the process model are developed. After functional testing the process automation software to the university portal are integrated via the user interface portlets.

The models are developed by using Eclipse BPMN modeler, in compliance with BPMN 2.0 standard. Activiti is utilized as the underlying process engine. Business rules are represented in Drools. All programs are coded with Java, JSP and JavaScript.

There are about 90 process modules identified in the business process map. The concept of "process module" is used for a group of process models which are coherent and focus on a specific working area of the organization (like budgeting). All of the process models under a process module are connected to the hierarchical structure of the related module. Until now, the team completed the development of 10 modules. Business process analysis and definition effort and total development effort are available for the modules completed.

In summary, there are two kinds of outputs for defining business processes: Process definition documents (named as "definitive process models" by the organization) and process models in BPMN (named as "analytical process models"). Process definition documents are organized in a tabular format and contain textual definitions of processes, detailed activity definitions, stakeholders, business rules, risks, inputs and outputs, objectives, entry and exit criteria of the process and data elements. Moreover, these documents also contain the requirements related to the automation of the process, and key performance indicators to be collected. Process models are developed using BPMN notation, however they are high level models representing only the control flow. The image of process models are also embedded in the process definition documents. In this case study, when the statement "native process definition document" is used, we refer to the existing documentation that is developed by the organization that contains all information of process, requirements, KPI and definitions.

The organization does not use any formal method for functional size estimation of the process modules to be automated. However, at every phase, they make an estimation of time and effort for each process module using similarity approach. They also collect effort and duration data for the development activities. This shows that they are willing to apply some formal approach to estimate the size of their modules. They also state that the organization needs to make effort estimation for the rest of the modules to be developed in phases to use as the basis of planning and budgeting.

The organization was willing to conduct different methodologies for their analysis activities and accepted to share information with us and to allocate personnel to study together in the research activities and evaluate the results. Considering the scope of the project, METU-IIS project conformed to all our criteria for our case study. We could conduct a case study that covers business process improvement, requirements engineering, software size estimation and process documentation practices. It is a retroactive type case study where we applied UPROM to obtain outcomes which were already developed in the study previously. We compared and evaluated the results with the previous results and discussed the findings. The organization agreed to work together and provided necessary effort to explain the system and evaluate the outputs together.

Among the process modules for which the process and requirements analysis was completed, we selected two process modules as the objects of this case study together with the organization. First process module is “Establishment of Research Opportunities” (will be called shortly as “IIS-ERO Module”. This was one of the process modules in research process area group, which was a high-importance process area for the organization which was planned to be fully automated. Second process module is “Announcement”, which was a central module utilized as a service by many other process modules. These two modules had mature process definitions. The automation of IIS-ERO was completed.

4.5.2. Objective

In order to answer the general research questions, this case study focused on integrating BPMod, requirements analysis, functional size estimation, process KPI identification and process documentation practices. Thus, to contribute to answering the research questions (provided in 4.1.2), the research objective of this case study was to examine, for METU-IIS project, if UPROM can be used to analyze and model processes, generate process documentation, develop process KPI list and user requirements and estimate the size of the software to be automated all together and based on business process models; and evaluate if these artifacts can cover the information provided in the existing process definition documents. Furthermore, improvements on the existing documents achieved by utilizing UPROM in consistency, completeness, maintainability and traceability is aimed to be evaluated.

4.5.3. Case Study Plan

The objective of this case study was to apply UPROM to generate the artifacts, collect and analyze data and answer the research questions. Different than the other case studies, outputs already developed in the project were used as inputs to the study as listed in the following section. The application of the case study started with a meeting with the organization to present UPROM and obtain their ideas. During model development, we contacted the organization's analysts to clarify the points when needed. At the end, meetings with the analysts were conducted again to evaluate the coverage of the generated artifacts, approve enhancements and obtain opinions for the methodology and the artifacts. Case study plan specific to this project is provided in Table 22.

Analysis activities for this case study included the comparison of results with respect to the existing outputs. Unlike the previous case studies, summary of observations and the result of analysis were provided in the last section of the "Case Study Conduct" for all artifacts. This is because METU IIS project covers the process definition, requirements and KPIs in a single document, native process definition document. The analysis of coverage for this documentation was conducted by comparing it to all artifacts generated by UPROM. So, the results are also presented in an integrated way.

Table 22 Case study plan for PublicInvest project

No	Activity	Inputs	Data to be collected	Data Collection Method	Analysis
1	Conduct a meeting with organization's analysts to introduce UPROM				Decide the process modules to be developed in the case study Obtain ideas on UPROM Learn details of the organization's implementation
2	Conduct descriptive business process analysis Add process meta-data, description, terminology, improvements, KPIs	Native process definition document Information obtained from the analysts	# processes # nodes, connectors, functions and arcs # diagram types	Metric collection	
3	Conduct function allocation analysis	Native process definition document Information obtained from the analysts			
4	Generate requirements document	Business process models Requirements in existing document	# total functional requirements # total requirements in existing documents		

5	Generate size estimation report Estimate the effort based on existing data sets. Compare the estimated and realized efforts.	Business process models Realized effort information from the organization	FP values measured by UPROM Effort estimation based on estimated FP size and realized efforts Evaluation of UPROM size measurement results	Observation Interview	Compare estimated and realized effort estimation
6	Generate process definition document Generate business glossary Generate process improvement list Generate process KPI list	Definitions from existing documentation and knowledge obtained from analysts Business process models			
7	Analyze the extent of information coverage for native documents by means of models and generated artifacts. Identify improvements on the existing information.	All outputs and analysis results from previous activities Native process definition document	% information coverage # proposed improvements		
8	Obtain approval for the information coverage table Obtain approval for process improvement list items Analyze the understandability of the models and artifacts and other benefits from the viewpoint of organization's analysts	Information coverage table Process improvement list	% information coverage (agreed) # improvements (agreed) Evaluation of maintainability and traceability Result of understandability and usability evaluation by users	Interview Observation Metric collection	Analyze the outputs as a whole to answer the research questions Identify the needs required but not covered with UPROM Identify improvements

4.5.4. Sources of Evidences

The objects of this case study are two process modules within the METU-IIS project: IIS-ERA and Announcement modules. The subjects of this study from which data is collected are the analysts from the organization. The sources of evidences to be used and collected during this case study can be summarized as follows:

Documentation: Outputs already developed in the project were used as inputs to make comparison with the case study outputs. The existing documentation were as follows.

- Native process definition documents: in a standard template, including most of the information we need while we develop process models conforming to UPROM. The content of the document included: textual definitions of processes, detailed activity definitions, stakeholders, business rules, risks, inputs and outputs, objectives, entry and exit criteria of the process and data elements, requirements related to the automation of the process, and KPIs, business process model images embedded.
- Business process models: BPMN notation, high level. Can be used as a starting point for modeling activities and to understand the control flow of activities in process documents
- Realized effort for process module development

Interviews: Interviews were conducted to validate the coverage of generated artifacts with existing information, the applicability of the proposed improvements and evaluate the understandability and usability of the methodology and the artifacts. Coverage table and process improvement list report were used as an input to the interviews. There were no other additional questions identified.

Direct Observations: In this project, we made direct observations by comparing the results of the existing project with the outputs delivered by UPROM. The results were evaluated together with the analysts of the project to validate the observations.

Participant Observation: Conducting the project activities with UPROM methodology together with the analysts from the organization, many observations were conducted as participants in the study, by means of which we had a deep understanding of the implementations and collect feedbacks.

In this case study, we utilized metrics collection (such as percentage of coverage) for quantitative analysis, and observations and interviews for qualitative analysis.

4.5.5. Conduct of the Case Study, Data Collection and Analysis

The case study was initiated by a meeting with the organization's personnel conducting the analysis activities, which were four analysts and a consultant. UPROM methodology and the tool was introduced in the meeting. The two process modules, "IIS ERO" and "Announcement" were selected to be applied in this case study. The organization presented the tools they utilized and the outputs they prepared. We conducted studies with the analysts during the case study when additional information is needed.

Developing Business Process Models and Analysis Models

At the initial phase, business process models for IIS ERO (Establishment of Research Opportunities) and Announcement process modules of METU-IIS project were developed using UPROM. At this first step, business process models were developed (FTD, EPC and OC diagrams) as mentioned in the Activity 1 of the Case Study Plan. As an input, existing process models and process definition documents and information obtained from organization's analysts were utilized.

The structure of the project under the main project "BBS" (standing for METU-IIS) is shown in Figure 55. FADs are not shown for the sake of readability, but they are placed under the same folder with their related EPC diagram from which it is referenced as a sub-diagram. As seen in the figure, under the main project folder, there are four models:

- Arastirma.erd and Duyuru.erd: The first one provides the high level entities and their relations in the system for IIS ERA Module, and the second for Announcement Process Module (shown in Appendix MM of the technical report [207]).
- BBS.ftd: the process map file for the project. All the process models in one level lower were linked on this process map (at Appendix LL of the technical report [207]). There were functions referencing the processes of other processes within METU-IIS system. These were not within the scope of the case study, but modeled to provide an overview of related process modules.
- BBS.oc: Organization chart that included all organizational elements and their relations within the project.

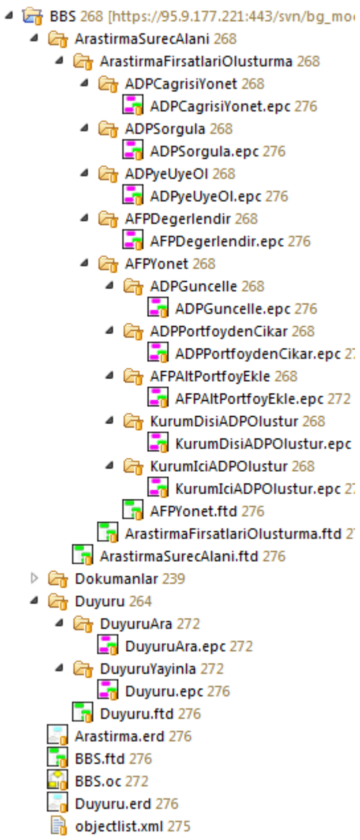


Figure 55 The modeling project structure of METU-IIS

Under the main project folder, there are sub-folders for the highest level processes in the project. The case study involved two process modules, thus the diagrams were detailed in two separate highest level folders. The names of the EPC models in order and in hierarchy are as follows:

- Research Process Area
    - Establishment of Research Opportunities
        - Manage Research Support Program Call
        - Search Research Support Program
        - Be a member of Research Support Program
        - Evaluate Research Opportunities Portfolio
        - Manage Research Opportunities Portfolio
            - Update Research Support Program
            - Remove Research Support Program from Portfolio
            - Add Sub-Portfolio to Research Opportunities Portfolio
            - Define External Research Support Program
            - Define Internal Research Support Program
- Announcement
    - Search Announcement
    - Publish Announcement

In the modeling of the processes, the following symbols were used:

- **EPC:** Event, Function, Process Interface, Application, Organizational Elements (Organizational Unit, Position), Document, Reference, Business Rule, KPI, Risk, Connectors (And, Or, XOR)
- **FTD:** Function
- **FAD:** Function, Entity, Application, Organizational Elements (Organizational Unit, Position, External Person), Constraint
- **ERD**: Entity, Relationship
- **OC**: Organizational Unit, Position, External Person

EPC diagrams were modeled in detail, providing alternative paths. The live validation functionality was utilized and the EPC models were developed free of semantic errors.

3 out of 11 EPCs had more than one start or end event. Because of this, we are aware that these diagrams are not sound. But because the focus was on business process analysis rather than formal analysis of the processes, multiple start and end events supported analysis activities.

FADs were developed for most of the functions on the EPC models, as most of the functions were to be automated by the system to be developed. All of the entities utilized in FAD diagrams were added to ERD diagrams so that entities and relations between them could also be observed as a whole and size measurement could be calculated precisely.

The structure of the project was established as explained in UPROM. The sub-folder structure corresponded to the sub-diagram hierarchy developed in the diagrams. Functional business process diagrams (FTD or EPC) were placed under a subfolder. The name of the folder and the name of the EPC/FTD diagram matched. On UPROM tool, project structure was validated successfully.

Data Collection – Metrics

The metrics for the developed processes are shown in Table 23. Detailed metrics for EPCs in METU-IIS are provided in Table 24.

Table 23 Metrics for METU-IIS business process models

Metric	**METU-IIS**	**Explanation**
The deepest level in the hierarchy	4	
Number of EPC diagrams	11	
Number of FTD diagrams	5	
Number of FAD diagrams	36	
Number of function symbols on EPCs	40	4 functions don't have FADs as they are not to be automated
Number of function nodes per EPC	3,6	
Number of FAD per EPC	3,3	
Total number of unique control flow nodes	145	Includes functions, connectors, process interfaces and events
Total number of unique control flow nodes without event nodes	75	
Average number of total control flow nodes per EPC diagram (without events)	6,8	
Average number of connector nodes per EPC diagram	2,1	
Average number of arcs per EPC diagram	12,9	

Table 24 Detailed metrics for EPCs in METU-IIS

	Function	Connector	Prc.Int. (instance)	Control flow nodes w/o events	Event	Control flow nodes	App.	Org. El.	Inf.Car.	Bus. Rule	Total	# of arcs	Start Event	End Event	KPI	Risk
ADPCagrisiYonet	2	2	1	5	6	11	1	1	0	2	15	10	3	1	2	1
ADPSorgula	2	0	0	2	2	4	1	1	0	2	8	3	1	1		
ADPyeUyeOl.epc	2	0	0	2	2	4	1	1	0	1	7	3	1	1		
AFPDegerlendir	2	2	5	9	12	21	1	1	0	3	26	20	2	5	3	
ADPGuncelle	2	0	1	3	3	6	1	1	0	2	10	5	1	1		
ADPPortfoydenCikar	1	0	1	2	3	5	1	1	0	2	9	4	1	1		
AFPAltPortfoyEkle	3	0	0	3	2	5	1	2	1	1	10	4	1	1		
KurumDisiADPOlustur	4	4	2	10	9	19	1	2	0	1	23	18	1	2	2	
KurumIciADPOlustur	4	2	2	8	6	14	1	5	1	3	24	14	1	1	1	
DuyuruAra	2	0	0	2	2	4	1	1	0	1	7	3	1	1		
Duyuru	16	13	0	29	23	52	1	3	17	0	73	58	6	3	5	1
Total	40	23	12	75	70	145	11	19	19	18	212	142				
Average	3,6	2,1	1,1	6,8	6,4	13,2	1,0	1,7	1,7	1,6	19,3	12,9				

As observed, average number of all nodes, function and connector nodes were less than the suggested values by "7 Modeling Guidelines" [9] as described in first case study. None of the EPCs exceeded the number of all nodes limit and one of them exceeded the number of connectors and arcs limit. In the overall, we can say that the models complied with the 7PMG for the size of the models (for G1, G2 and G7).

In METU-IIS project, 3 processes had multiple start and end events. Special focus was given to match start and end events in UPROM so that the flow was smooth between the main diagram, sub-diagrams and process interfaces. We think that multiple start and end events enhanced expressiveness of the model in this case study and decreased error probability as total number of nodes and similar activities decrease by means of grouping the activities for multiple start and end events.

All splits and joins were paired apart from the ones used to indicate multiple close and end events, and the joins that indicated feedback returns (that are used to indicate some part of the process to be restarted because of a feedback in process activities). Or connector was never used in METU-IIS models. The depth of the hierarchy is 4. The level of granularity for the functions was tried to be kept same throughout the projects. All of the activities were labeled in an imperative way, as suggested by the guideline (G6).

Generating Requirements Document

As mentioned in Activity 2 and 3 of the Case Study Plan, requirements for the system were analyzed conforming to UPROM. Native process definition documents were utilized as input to this analysis. These documents contained both process definitions and requirements applicable for the system. In these documents, some of the requirements sentences defined the process flow, rather than functional requirement of the system. These were utilized to define the business processes as in the previous step. After the completion of the analysis, functional requirements were generated for the system.

The generated "User Requirements Document" can be seen in Appendix NN of the technical report [207]. Three types of requirements sentences were generated:

Type 1: In this type of sentence, the organizational elements conducting the related function were stated. For some of the FADs, there was no organizational element connected to the function, as the function was automatically conducted by the system. Examples of such requirements sentences were also generated within this case study.

Type 2: This sentence type was generated to express the operations conducted on the entities and systems. Some of the requirements in the native process definition documents expressed the same content with this type of requirements.

Type 3: The sentence type was generated from constraints placed on FADs. Most of the constraints placed on FADs were based on the requirements sentences on native process definition document.

We made one small update to the requirement generation procedures of UPROM so that requirements sentences conformed to the existing sentences in the documents. If the name of the application included “service”, then the statement for “system” was omitted, and the sentence was formed based on the word “service”.

Data Collection – Metrics

The metrics for the generated requirements for METU-IIS are shown in Table 25.

Table 25 Metrics for METU-IIS requirements

Metric	**METU-IIS**
Total number of generated functional requirements	125
Total number of requirements generated from constraints (constraints on FADs)	53
# requirements in native process definition document	65 (46 for Announcement and 19 for IIS-ERO)
# requirements covered by the generated requirements document	61 (1 is quality requirement and 3 is evaluated to be not applicable, as they are requirements of other systems)
% of requirements coverage	94%
Number of EPC diagrams	11
Number of requirements per EPC diagram	11,4

Generating COSMIC Functional Size Estimation

As mentioned in Activity 4 of the case study plan, size estimation report was generated for the two process modules of METU-IIS project by means of UPROM tool, which generates the size estimation results automatically from business process models as explained in UPROM. There were only two applications in the process models: Announcement and IIS-ERA Module. Related processes did not provide or obtain information from external systems. Thus, size measurement activities only covered the estimation for the two applications. The generated “Size Measurement Report” can be seen in Appendix OO of the technical report [207].

Data Collection – Metrics

The estimated size for each module of METU-IIS is provided in Table 26.

Table 26 UPROM estimated size for software modules in METU-IIS

Module	**FP Size**
IIS ERA	110
Announcement Service	94
TOTAL	204

Table 27 Distribution of data movement types for METU-IIS

Entry	**Read**	**Write**	**Exit**
51	58	40	55
25%	28%	20%	27%

Considering the distribution of data movement types according to domains [209], [210], the result of size estimation provided in the above table is in harmony with the ISBSG findings for business application domain; having less number of write data movements and the rest being close to each other. Only the number of entries are higher than the averages. Considering this, we can claim that UPROM size estimation produced reasonable data movement weights for a business application domain system.

Generating Process Documentation

Within the scope of UPROM, process documentation includes four artifacts: Process definition document, business glossary, process improvement list and process KPI list. Additional terminology definitions, process descriptions, descriptions for the functions and KPI's were already defined on the models during Activity 2. Also, the improvements were identified during modeling. Process documentation were generated by means of UPROM tool generation functions.

The generated "process definition document" can be seen for METU-IIS in Appendix PP, business glossary can be seen in Appendix QQ, process improvement list can be seen in Appendix RR, and process KPI list can be seen in Appendix SS of the technical report [207]. An excerpt from the process KPI list is provided in Figure 56.

Ölçüm	Açıklama	Ölçüldüğü Aktivite	Ölçüldüğü Olay	Kullanılan Bilgi Kaynakları	Ölçme Zaman ve Sıklığı	Hedef
Kullanılan Şablon Sayısı	Şablon türü başına kullanma sayısı	Duyuru şablonunu seç		duyuru şablon türü	Ölçüm dönemi	
Şablon doldurma süresi	Şablon türü başına doldurma süresi (en kısa, en uzun, ortalama)	Duyuru hazırla		duyuru (tam dolu)	Ölçüm dönemi	
Onay gerektiren duyurular için duyurunun yayınlanma süresi	Onay gerektiren duyurular için, duyuru şablonunun doldurulması ile yayınlanması arasında geçen süre (en kısa, en uzun, ortalama)	Duyuruyu elektronik kanallarda yayımla		duyuru (tam dolu) duyuru (yayımlanıyor)	Ölçüm dönemi	1 hafta

Figure 56 Excerpt from METU-IIS process KPI list

Data Collection – Metrics

The metrics on the process definition documents, business glossary, process improvement list and process KPI list outputs are provided in Table 28.

Table 28 Metrics on process documents for METU-IIS

Metric	METU-IIS
% of process models for which general process descriptions (process metadata) are added	100%
% of functions for which descriptions are added (except the ones referencing sub-processes)	83%
Number of entries in business glossary (includes the ones which don't have technical term description but added because it is a part of a high level entity)	93
Ratio of business glossary entries per total number of entities in the project	100% (93/93)
Number of business glossary entries per EPC diagram	8,5
Number of improvements	64
Number of improvements per EPC diagram	5,8
Number of KPIs	11

We observe that for all of the process models, general process descriptions were added as attributes. Also, many of the functions were detailed with additional descriptions. We can see that attributes for process models and descriptions for the functions were helpful to reflect the information provided in the existing documentation.

We added all of the findings to improve the information on the existing process documentation as improvement opportunities on the process models. This was done during the modeling process. That is the reason for high number of improvements.

The KPIs were added as specified in the existing documentation. No additional KPIs were added apart from the ones already defined. However, improvements were proposed for existing KPIs, and removal of some KPIs were suggested as they were not related to the existing processes.

Evaluating the Overall Results

Data Collection – Metrics

Upon completion of the process modeling activities and generation of the artifacts within the case study, we analyzed each section in the existing process definition document for the two process modules for coverage. Each section and information in the document (named as items) was mapped to UPROM artifacts. For this analysis, for each section, we identified the following information:

- Section and information in the existing process definition document
- How we added this information on UPROM models

- On what UPROM artifact (and part of the artifact) this information resides
- Is the mapping native (straightforward) or adapted (the information can be somehow expressed but not as a major functionality of UPROM)
- If the information is not covered, can it be added? (and how)
- Are there any excess information or benefit achieved by expressing the related information using UPROM?

The analysis for coverage of information in native IIS process definition documents with UPROM processes and artifacts is provided in Appendix SS of the technical report [207]. The metrics for the coverage analysis are provided in Table 29 below.

Table 29 Metrics for coverage analysis of METU-IIS project with UPROM artifacts

Metric	**IIS ERA Process**	**Announcement Process**
# items in the native process definition document	59	125
#items covered by UPROM artifacts in native way	46	118
# items covered by UPROM artifacts in adapted way	6	3
# items identified as not applicable (N/A)	4	1
# items not covered by UPROM	3	3
# items adapted or not covered but that can be added to UPROM	7	5
# items for which enhancements are identified by expressing the information in UPROM	32	45

In addition to the coverage analysis, the improvements introduced by following UPROM as the methodology to analyze business processes, requirements and generate the artifacts were identified as improvement opportunity symbols on the models. The process improvement list was generated as an artifact. Total number of improvements identified in each diagram and the type of the improvement is shown in Table 30 and Table 31 below.

Table 30 Improvements introduced by applying UPROM organized by EPC diagram

Process Module	**Diagram**	**Number**
Announcement	Publish Announcement	35
	Search Announcement	1
	Total	**36**
IIS ERO	Manage Research Support Program Call	3
	Search Research Support Program	2
	Be a member of Research Support Program	2
	Evaluate Research Opportunities Portfolio	3
	Update Research Support Program	3
	Remove Research Support Program from Portfolio	2
	Add Sub-Portfolio to Research Opportunities Portfolio	4
	Define External Research Support Program	5
	Define Internal Research Support Program	4
	Total	**28**
Total		**64**

Table 31 Improvements introduced by applying UPROM organized by type

Improvement Type	**Number**
Process modeling	29
Requirements	22
Unified definition and descriptions	9
KPI	4
Total	**64**

Data Collection – Summary of Interviews

The meetings for METU-IIS project were conducted with 4 analysts from the organization. We consulted the organization's analysts whenever required throughout the modeling and at the end, presented them business process models, generated artifacts, coverage analysis and improvement opportunities. We evaluated each item in the coverage analysis and the improvement opportunity list; updated the results based on their opinions and obtained their approval. During this meeting, ideas on UPROM are also obtained from them.

- Usage of UPROM to generate artifacts that cover the information identified in the project.

The results of the coverage analysis depicted that only 3% of the information on the existing documentation was left fully uncovered for the two process modules (6 over 184). 5% of the information was covered in an adapted way, meaning that UPROM did not support the modeling of the related information in a direct way, but the information could be added to the models. An example to such an item is the code for the process model. A separate meta-data did not exist to store this information, but it could be added for example, as part of the model name metadata. If required, specific metadata for such an item can also be added. Only less than 1.6% of the items were not covered by UPROM and cannot be added to the methodology in a straightforward way.

As a result, we observe that UPROM could be used to analyze the business processes and develop business process models, requirements document, process definition document, business glossary and process KPI list in METU-IIS project so that all information in the existing process definition documents could be captured by UPROM artifacts.

In addition to these artifacts, UPROM was used to estimate the size of the related software applications. Size estimation in COSMIC FP was utilized to estimate the effort using the coefficients utilized in e-LegalEntity and e-Trademark case study. The resulting effort estimates are shown in Table 32. The productivity ratio in the table were achieved based on industrial values [211]. We believe that productivity ratio must be calibrated by the organization considering organizational and project properties. Still, the estimated effort for IIS-ERO was very close to the realized effort for this module, as reported in a previous study [212]. The development effort of the announcement module was not complete yet. The analysts of the organization stated that achieving an estimate of the software size automatically from the models is a good idea and they had never encountered such an approach before. They already had a need for estimating the size and effort for the modules by using process models and developed an effort prediction model for that [212]. They planned to incorporate the estimated COSMIC FP into this effort prediction model.

Table 32 Estimated effort for METU-IIS modules

Application	**Estimated COSMIC FP**	**Productivity Ratio (effort/FP)**	**Estimated Effort (person-hour)**	**Estimated effort (requirements analysis effort decreased)**
IIS-ERO	110	0,031	3,41	3,1
Announcement	94	0,031	2,92	2,7

The other artifact developed by UPROM but did not exist in the existing documentation was process improvement list. This functionality was utilized during the case study to identify the improvements introduced by following UPROM. Thus, the generated improvement opportunity list was not an output for METU-IIS project. However, organization's analysts agreed that noting down the improvements during process modeling is an effective way to identify improvements. And the resulting report was beneficial to make the benefits obtained visible and present to higher level management.

The organization stated that they already planned to analyze the models in an integrated way in their existing studies and utilized portals and common definition pools to assure the information is gathered and reached from one storage. They didn't store all the information on process models but stored them as separate groups of information. Thus, they indicated that as UPROM provides an environment to capture all required information on the models, it enhanced maintainability and traceability of all generated artifacts to the process models.

As a result, we can conclude that as an answer to the first research question, we were able to use UPROM to analyze and model processes, generate process documentation, develop requirements and estimate software size in an integrated way based on business process models. These artifacts were easily maintainable and traceable to business processes. The result of coverage analysis showed that the generated artifacts covered most of the information provided in native process definition document.

- The effect of UPROM to provide enhancement and ensure accuracy, consistency and completeness

184 items were identified on the existing process definition documents for the two process modules analyzed, and in 77 of these items, enhancements are identified. These are shown in Table 30. Additionally, 92 improvements were identified in total in the process improvement list. Further analyzing the improvements, we observed that they

were identified for the accuracy, consistency and completeness problems in different categories, as shown in Table 31. Thus as an answer to case study research question 2, we conclude that UPROM provided enhanced expression of the information already captured by the process definition documents, and introduced further improvements to improve accuracy, consistency and completeness of the process definitions, requirements, business glossary definitions and KPI definitions.

Wrapping up these results, we answer the case study's research questions as follows.

CSQ1: UPROM methodology and the tool were used for METU-IIS project to analyze and model the processes, develop user requirements, estimate the size of the software and generate process definition document, business glossary, process improvement list and process KPI list. All of the knowledge in native process definition document was captured on process models and related artifacts were generated automatically. The resulting artifacts were able to cover 97% of the information in native process definition document. Additionally, software FSE, which is not an output included in the project, was achieved without further effort.

CSQ2: Enhancements were achieved by following UPROM in the project. Enhancements on accuracy, consistency and completeness of the artifacts were identified in 42% of the items in native process definition document. 92 additional improvements were identified.

4.6. Overall Results and Threats to Validity

Overall Results:

As described in the previous sections, multiple case study was applied as the research methodology to evaluate UPROM. Considering the results of case studies with respect to the research questions, we achieved the following results.

We applied four case studies, two of them covering the aspects of UPROM methodology partially and two of them involving multiple cases. Three of these case studies were conducted in real life settings and results were presented to the customers. In cases 2 and 3, case study outputs were delivered to the customer and passed the formal acceptance phase as project deliverables. In the fourth case study, outcomes of a project for which business process and requirements analysis were already conducted were almost fully covered by artifacts generated by UPROM and many enhancements were identified. Considering these aspects for these cases, we conclude that we could use UPROM to develop business process models, obtain requirements and early software size estimation and process documentation in an integrated way. To provide information on the outputs developed in the case studies and depict characteristics of the cases, an overview of case study metrics is provided in Table 33.

Table 33 Metrics summarizing the outputs of case studies

Case Study	Project	# EPC, FTD & VC diagrams	#Functions	#FADs	# Generated requirements	#Manually added requirements	Estimated Size as COSMIC FP	Process definition doc. page #	# Business glossary entries	# Improvements	# KPIs
Case Study 1	Celebrity Management	5	12	8	21	-	38	-	-	-	-
	Veterinary Record System	4	11	7	19	-	36	-	-	-	-
	Movie Management	3	13	11	28	-	83	-	-	-	-
	Total	10	36	26	68		157				
Case Study 2	e-Company	18	125	82	363	35	1102	61	82	18	-
	e-Trademark	9	47	36	177	19	437	28	40	7	-
	Total	27	172	118	540	54	1539	89	122	25	
Case Study 3	Public Investments	10	65	-	-	-	-	34	93	19	-
Case Study 4	IIS-ERO	13	42	18	60		110	33	28	28	6
	Announcement	3	25	18	65		94	11	28	36	5
	Total	16	67	36	125	4	204	44	56	64	11

Considering the metrics, we observe that the cases had different characteristics in terms of size, complexity, level of detail and purpose. In addition to the metrics collected for the cases, data collected from observations and interviews were utilized to evaluate the results of the multiple case study. Having developed all of these outputs utilizing the same source of information, business process models, and considering the analysis of results from the case studies described in the previous chapters, we conclude that we could achieve the following benefits. These results were supported by observations and interviews. Wrapping up all evidences collected from the multiple cases, we can answer the research questions in the following way:

RQ1: We developed and utilized UPROM as the methodology to analyze and model the processes; develop user requirements (cases 2 and 4); estimate the size of software to be automated (case 1, 2 and 4); generate process definition document, business glossary and process improvement list (case 2, 3 and 4), and process KPI list (case 4). In all of the cases, process models were utilized as the single source of information and could be generated automatically.

RQ2: For the generated artifacts, based on the observations, metrics and interviews, we conclude that enhancements were achieved as follows.

- Business process models were developed in a well-structured way, as the models were developed conforming to UPROM guidelines and guidelines suggested in the literature. As revealed by the interviews, the guidelines supported process analysis activities and development of understandable models.
- User requirements were obtained which accurately described the system to be automated and provided a complete description of the system in a consistent manner. In Case 2, the requirements composed 90% of the technical contract, and 10% included general aspects of the system. Thus, the requirements were complete in terms of the business processes of the system. Analyzing the requirements with this method provided an easy and standardized way to develop requirements in a complete way.
- Software size could be estimated in early phase of the projects with reasonably accurate results compared to measurement. It proved to be helpful to plan the automation project in an early phase, and without spending any extra effort for measurement other than analysis conducted already for requirements.
- Process documentation including process definition document, business glossary, process KPI list and process improvement list was generated in a complete and consistent manner. Process definition document, business glossary and process KPI list outputs were utilized to provide understanding of the processes for different stakeholders and express information on the processes in a standardized way which were not easy to state with process models. Process improvement list was used to report the benefits obtained by the processes.
- All of these outputs were easy to maintain in case of any changes in the processes and the system. When updates were needed, changes were conducted only on the process models and artifacts were regenerated automatically. Stakeholders in the case studies agreed that this provided high maintainability for the artifacts.
- All artifacts were traceable to process models.

In general, we can conclude that the artifacts generated from the models were complete with respect to business process models. We do not deduct a general claim that all artifacts that will be required in early analysis phases of similar projects can be produced by applying UPROM. Additional needs can emerge due to project needs. Also, we do not claim that the information required for all similar artifacts can always be developed in a complete way. For example, in Case 2, additional requirement statements were added to the technical contract. However, we claim that UPROM provides the modelers an integrated environment to capture information for the related practices within the models. Thus, modelers can achieve a complete set of artifacts with regard to business process models. The completeness of the requirements for the functions in business process models, the items in the process definition document, business glossary, improvements and KPIs were observed to be enhanced within the scope of business process models. As a result, the quality and coverage of business process models affected the completeness of the related artifacts.

UPROM has an impact for consistency of the artifacts in both internal and external level. In this study, we focused on enhancing internal consistency of artifacts by conducting the required analysis and generating them based on business processes. It is important to model disjoint and modular set of business processes, for which guidelines are provided in this study. If models are disjoint, as the analysis activities are steered by those business processes, information for other practices is also captured distinctly. This enhances the internal consistency of the artifacts. Moreover, we observed that external consistency, consistency between the artifacts, were also enhanced as all of the artifacts were based on the same source. The major source of redundancy and inconsistency is independent development and update of artifacts. By utilizing UPROM, the information required for all artifacts are captured in parallel and impacts of changes are analyzed on the single source. By means of this aspect, redundancy of the captured information as a whole was decreased external consistency of the artifacts was also improved.

Threats to Validity

As a result of the application of case study research, some possible threats to validity arises. During the planning phase of the case studies, actions were planned to overcome these threats. Here we explain, for each threat, the actions conducted to avoid the threats and the situation.

The construct validity may be a problem if the case study activities are not appropriate to evaluate the method, metrics collected and observations are not interpreted in a correct way and interviews are not conducted in a way to reach correct results to answer the research questions [201], [203]. To avoid these problems, we had the case study plan and interview questions reviewed by multiple external experts before conducting the case studies and discussed how to analyze the outputs with respect to research questions.

Internal validity is our concern as we try to make conclusions on the outputs derived by means of applying the methodology. Application of multiple case studies is especially important to overcome this threat. The outputs delivered as a result of applying the methodology were already to be delivered as a result of standard business process and requirements analysis activities. Thus, the quality of them may be affected not only by the application of UPROM methodology, but by also various conditions. To avoid this risk and reveal how the outputs are affected because of the method application, a chain of evidence was maintained while conducting the study and reporting the results. The evidences created by the case studies were provided in detail in the technical report [207] and referenced from the relevant points in this case study report. The background on the cases, the objectives of the related studies (independent from the case study application) and how those studies utilized the case study outputs in real life setting were clearly defined. For example, the requirements document was used as the technical contract and used in project bidding in e-LegalEntity project, whereas improvement list was provided only as an example to the customer and evaluated accordingly. Different sources of data were utilized to analyze the results and answer the research questions as a whole. Metrics on different aspects of the outputs were collected, evaluated and compared with relevant data when applicable. The resulting outputs were validated by formal acceptance procedures of the projects in real life settings and conducting interviews with the related stakeholders. Also, a retrospective case study was applied where the results achieved by UPROM were compared to the ones previously developed with other techniques in the organization. The stakeholders from the organization reviewed and approved the comparison results.

External validity deals with the concern of the generalizability of the results of the case study [203]. In case studies, rather than statistical generalization, we try to find analytical generalization of the results to some broader theory [201]. We overcame this threat by specifying the conditions for applicability of UPROM and conducting multiple case studies where we can apply replication logic. The conditions for which UPROM is applicable were defined, an organization requires to analyze business processes and automate them, as explained in detail in section 4.1.1. Also, we applied UPROM in service based organizations, applicability in manufacturing and other types would require more examination. We ensured that we applied the replication logic consistently through the cases by means of UPROM tool as it guides to implement the methodology. We ensured that the tool works and generates outputs as described by the methodology by means of the experts which reviewed the outputs and compared with manually generated ones. These reviews also supported the internal validity, as it ensured that consistent outputs could be delivered through multiple executions of the same or different cases.

Many actions were taken to prevent reliability problems and ensure that other researchers can conduct the same study following the methodology. Firstly, a case study protocol was followed and explained in detail in for each case study, where the case is introduced, case study plan is described, sources of evidences are identified and the analysis methods are defined. A case study database was formed and all outputs were provided within reach of other researchers. Moreover, by assuring that UPROM tool works as described by the methodology, researchers even did not need to apply the methodology manually but just used the tool to replicate the results and end up with the deliverables. All models created and outputs generated, which could not be placed in this document for size concerns, were made available by means of a technical report.

To ensure the reliability of UPROM tool to work as specified by UPROM, external expert reviews were planned and conducted on each artifact of the methodology. In case study 1, FSE report was evaluated by an external expert and ensured that size estimation values were calculated as specified by UPROM FSE method. In case study 2, requirements, process definition document, business glossary and process improvement list was checked using business process models on the tool. It was confirmed by an external expert that all artifacts were generated in conformance with artifact generation procedures. Moreover, as the projects of case study 2 and 3 were conducted in real life settings, project stakeholders reviewed and approved the artifacts as project deliverables.

CHAPTER 5

CONCLUSION

Organizations analyze and model their business processes for various purposes. In this study, we focus on utilizing the knowledge obtained in business process models to fulfill the needs of organizational and software practices. This thesis proposes a unified BPMod methodology, UPROM, to integrate many practices related to business process models. These practices are requirements analysis, software functional size estimation and process metrics definition for the process automation software; and process definition document and business glossary for process documentation. In this chapter, the contributions achieved by the proposed methodology are summarized and limitations and planned future work are presented.

5.1. Contributions

Throughout the conduct of this study, we focused on different practices for utilizing business process models, then to integrate all those approaches to define the complete methodology. We have published several papers as a result of these studies.

The major contribution achieved by this study is the unified BPMod methodology, UPROM designed to conduct BPMod while automatically generating artifacts for related practices. It is hard to find methodologies in BPMod field that guides users thoroughly. UPROM methodology provides a complete description of the activities to be conducted to achieve the benefits by covering a notation, meta-model, process, guidelines and artifact generation procedures. While achieving a unified approach for BPMod, different perspectives of an organization including functional, behavioral, organizational and informational are also represented in the models.

In order to validate applicability of UPROM, a multiple case study with four cases were planned and conducted. Each of these cases provided a different perspective to validate the methodology. The first case focused on software size estimation aspects, the second provided a complete application to deliver the outputs of the project, the third conducted activities for process documentation purposes in a different detail level, and the fourth provided comparison and evaluation of artifacts generated previously for a project in a retrospective style. UPROM could be used in a modular way when the organization does not require all practices within the scope of UPROM. Thus, we can conclude that UPROM can be used to meet the needs of studies with different needs and in different domains.

The case studies showed that UPROM can be used to conduct BPMod and generate the artifacts required in the projects. In all case studies, the resulting artifacts are developed in a more complete way with respect to business process models. Internal consistency of the artifacts were enhanced and redundancy in the captured information was decreased as the analysis were conducted on business processes. As all the artifacts were developed based on the business process models, they were traceable to the models. When there was a need to update models or any of the artifacts, the change was applied only on the models. Then, without the need for analyzing the effects of change for each artifact, all artifacts were generated again and up to date versions were achieved. In this way, external consistency between the artifacts were also enhanced and consistency problems between them due to independent development of artifacts were minimized.

Another important benefit obtained is the UPROM tool developed as a prototype to apply the methodology. The tool is utilized successfully in all case studies. It enabled the development of business process models in conformance with UPROM, and generation of artifacts in an automated way.

In addition to the overall benefits obtained by integrating all practices, UPROM provides benefits for each practice it focuses on. By generating requirements document, the following benefits were observed:

- Requirements are defined to fit better to the criteria identified by IEEE requirements standards [128] to be unambiguous, complete, consistent and uniquely identifiable.
- The models provide better analysis of requirements by analysts, while textual statements enable end users to better communicate and validate them.
- An automatic traceability between business processes and requirements statements are provided.
- The required effort to write requirements statements may not be less in the initial definition phase compared to manual methods. But if the maintenance effort is considered, effort savings will be achieved in the long run.

By generating software functional size estimation, the following benefits were observed:

- Functional size estimation of the software is obtained in an early phase compared to traditional functional size measurement activities.
- Reasonably accurate functional size estimation is achieved, thus reliable effort estimation and project planning can be conducted.
- Functional size estimation is achieved with almost no effort, as the estimation is automated by the tool.
- Estimation can be repeated easily and maintenance effort is minimized.

- Variation in measurement and subjectivity due to the measurer are decreased by using models and automating the estimation.
- Training needs for measurement are eliminated due to automation.

By generating process KPI list, the following benefits were observed:

- A systematic process evaluation system is established for the organizations.
- The relation of metrics to evaluate the performance are defined in association with the processes.
- The collection and calculation procedures are clarified so that KPI definitions can be used as a requirement for and easily be implemented in process automation software.
- Update needs for KPIs become visible when processes are updated.

By generating process documentation including process definition document, business glossary and process improvement list, the following benefits were observed:

- A document in natural language which is required by different types of stakeholders which do not prefer process models or prefer documents together with models is prepared.
- Process definition document is assured to conform to a certain structure and cover the required fields in all headings, as is usually a problem with the documents prepared manually.
- Details that are hard to express in the models or cause the models to be complex are expressed.
- Details on the relations between the processes are provided.
- Unique and complete definitions of concepts used in the business processes are achieved by means of business glossary.
- Improvements identified during BPMod can easily be recorded and obtained as a report.
- Process documentation is achieved with almost no effort, as the generation is automated by the tool.
- In case of changes, it is easy to observe its effects on the models. As all documents are generated again, documents remain consistent and up to date.

The benefits obtained by UPROM become more visible particularly if the organization plans to use it for multiple practices. Effort savings will be significant, as all the savings will be added up for each practice.

5.2. Limitations and Future Work

An important limitation of case study research is generalization. We observed the application of UPROM in different organization types and needs, still the number of the studies must be increased to assure its applicability for different conditions. The methodology is focused and the applications were conducted on service based organizations. The methodology does not aim to provide a solution for manufacturing or other types of organizations. Further studies need to be conducted to evaluate its applicability for those organization types.

The benefits obtained by UPROM can be thoroughly confirmed only after process automation software development is completed, software is deployed and users start using the system. However, achieving such a state requires a very long time. Moreover, it may not be possible to collect related data, like realized effort, in the organizations.

We believe that many practices can be further integrated within such an approach. One of the examples is software testing. Moreover, we identified many improvement opportunities regarding the details of the methodology. A list is provided below.

- Generation of requirements sentences in English.
- Integration of BPMN to the notation as alternative notation to EPC.
- Modeling for architectural design of applications.
- Analysis of internal and external entities and information carriers.
- Categorization and further guidelines to identify business rules.
- A method to define requirements general to the system on the models.
- Documentation like roles and responsibilities list.
- A plugin for the tool to design the format and content of the artifacts.

REFERENCES

[1] O. Turetken and O. Demirors, "Process modeling by process owners: A decentralized approach," *Softw. Process Improv. Pract.*, vol. 13, no. 1, pp. 75–87, 2008.

[2] J. Becker, M. Rosemann, and C. Von Uthmann, "Guidelines of Business Process Modeling," in *Business Process Management (2000)*, 2000, pp. 30–49.

[3] M. A. Ould, *Business Processes : Modelling and Analysis for Re-Engineering and Improvement*, 1st ed. Wiley, 1995.

[4] O. Demirors and F. Celik, "Process modeling methodologies for improvement and automation," in *Quality and Reliability (ICQR), 2011 IEEE International Conference on*, 2011, pp. 312–316.

[5] M. Dumas, W. Van der Aalst, and A. ter Hofstede, *Process-aware information systems : bridging people and software through process technology*. New Jersey: John Wiley & Sons, 2005, p. 426.

[6] M. Weske, *Business Process Management: Concepts, Languages, Architectures*, 2nd ed. Springer Publishing Company, Incorporated, 2012.

[7] F. Garcia, A. Vizcaino, and C. Ebert, "Process Management Tools," *IEEE Softw.*, vol. 28, no. 2, pp. 15–18, 2011.

[8] F. Çelik and O. Demirörs, "Utilizing Process Definitions for Process Automation : A Comparative Study," in *2010 BPM and Workflow Handbook: Spotlight on Business Intelligence*, L. Fischer, Ed. Future Strategies, Incorporated, 2010, pp. 247–256.

[9] J. Mendling, H. A. Reijers, and W. M. P. van der Aalst, "Seven process modeling guidelines (7PMG)," *Inf. Softw. Technol.*, vol. 52, no. 2, pp. 127–136, Feb. 2010.

[10] T. Gulledge, "Integrated Business Process and Service Management," in *Handbook on Business Process Management 1 SE - 22*, J. Brocke and M. Rosemann, Eds. Springer Berlin Heidelberg, 2010, pp. 481–496.

[11] W. M. P. Van Der Aalst, "Process-Aware Information Systems: Lessons to Be Learned from Process Mining," in *Transactions on Petri Nets and Other Models of Concurrency II*, K. Jensen and W. M. Aalst, Eds. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 1–26.

[12] M. Indulska, P. Green, J. Recker, and M. Rosemann, “Business Process Modeling: Perceived Benefits,” in *Conceptual Modeling - ER 2009 SE - 34*, vol. 5829, A. F. Laender, S. Castano, U. Dayal, F. Casati, and J. Oliveira, Eds. Springer Berlin Heidelberg, 2009, pp. 458–471.

[13] P. Jamshidi and C. Pahl, “Business process and software architecture model co-evolution patterns,” in *Modeling in Software Engineering (MISE), 2012 ICSE Workshop on*, 2012, pp. 91–97.

[14] H. C. Mayr, C. Kop, and D. Esberger, “Business Process Modeling and Requirements Modeling,” in *Digital Society, 2007. ICDS ’07. First International Conference on the*, 2007, p. 8.

[15] J. Vom Brocke and M. Rosemann, *Handbook on business process management 1*. Springer-Verlag, 2010, p. 632.

[16] J. Nicolás and A. Toval, “On the generation of requirements specifications from software engineering models: A systematic literature review,” *Inf. Softw. Technol.*, vol. 51, no. 9, pp. 1291–1307, 2009.

[17] K. Cox, K. T. Phalp, S. J. Bleistein, and J. M. Verner, “Deriving requirements from process models via the problem frames approach,” *Inf. Softw. Technol.*, vol. 47, no. 5, pp. 319–337, 2005.

[18] O. Turetken, O. Su, and O. Demirors, “Automating software requirements generation from business process models,” in *Proceedings of 1st Conf. On the Principles of Software Eng. (PRISE’04)*, 2004.

[19] A. Coskuncay, B. Aysolmaz, O. Demirors, O. Bilen, and I. Dogan, “Bridging The Gap Between Business Process Modeling And Software Requirements Analysis: A Case Study,” *MCIS 2010 Proc.*, p. Paper 20, 2010.

[20] L. Santillo and R. Meli, “Early Function Points: some practical experiences of use,” in *ESCOM-ENCRESS ’98 Conference Proceedings*, 1998.

[21] F. Vogelezang, “Early estimating using COSMIC-FFP,” in *SMEF Software Measurement European Forum 2005*, 2005, pp. 227–234.

[22] M. Kaya and O. Demirors, “E-Cosmic: A Business Process Model Based Functional Size Estimation Approach,” in *Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on*, 2011, pp. 404–410.

[23] C. Monsalve, A. Abran, and A. April, “Measuring Software Functional Size From Business Process Models,” *Int. J. Softw. Eng. Knowl. Eng.*, vol. 21, no. 03, pp. 311–338, May 2011.

[24] “ISO 9001:2008,” 2008.

[25] “CMMI® for Development, Version 1.3 CMMI-DEV, V1.3,” 2010.

[26] *Control Objectives for Information and Related Technology (COBIT)*. ISACA, 2010.

[27] K. Brennan, *A guide to the Business Analysis Body of Knowledge (BABOK guide), version 2.0*, 2nd Editio. IIBA International Institute of Business Analysis, 2009.

[28] V. Paradigm, "Develop as-is and to-be business process," 2012. [Online]. Available: http://www.visual-paradigm.com/tutorials/as-is-to-be-business-process.jsp. [Accessed: 30-Apr-2014].

[29] R.-H. Eid-Sabbagh, M. Hewelt, and M. Weske, "A Tool for Business Process Architecture Analysis," in *Service-Oriented Computing SE - 61*, vol. 8274, S. Basu, C. Pautasso, L. Zhang, and X. Fu, Eds. Springer Berlin Heidelberg, 2013, pp. 688–691.

[30] E. Aydin, O. Demirors, and O. Bilen, "İş süreci modellerinden kalite el kitabı üretmek için bir araç," in *Yazılım Kalitesi ve Yazılım Geliştirme Araçları Sempozyumu*, 2010.

[31] "5018 Sayılı Kamu Mali Yönetimi ve Kontrol Kanunu," *Resmi Gazete Sayı 25326, Tertip 5*, 10-Dec-2003.

[32] P. Kung, C. Hagen, M. Rodel, and S. Seifert, "Business process monitoring measurement in a large bank: challenges and selected approaches," in *Database and Expert Systems Applications, 2005. Proceedings. Sixteenth International Workshop on*, 2005, pp. 955–961.

[33] B. Aysolmaz, A. Coskuncay, O. Demirörs, and A. Yildiz, "Kamuda İş Süreçleri Modelleme: Gereği ve Yararları," in *5. Ulusal Yazılım Mühendisliği Sempozyumu - UYMS'11*, 2011.

[34] J. Recker, M. Rosemann, M. Indulska, and P. Green, "Business process modeling: a comparative analysis," *J. Assoc. Inf. Syst.*, vol. 10, no. 4, pp. 333–363, 2009.

[35] U. D. of the Treasury, "Treasury Enterprise Architecture Framework (TEAF)," 2000.

[36] W. Sedera, G. Gable, M. Rosemann, and R. Smyth, "A success model for business process modeling: findings from a multiple case study," in *Proceedings Eighth Pacific Asia Conference on Information Systems*, 2004, pp. 485–498.

[37] B. Curtis, M. I. Kellner, and J. Over, "Process Modeling," *Commun. ACM*, vol. 35, no. 9, pp. 75–90, 1992.

[38] C. Thörn and T. Gustafsson, "Uptake of Modeling Practices in SMES: Initial Results from an Industrial Survey," in *Proceedings of the 2008 International Workshop on Models in Software Engineering*, 2008, pp. 21–26.

[39] F. Celik, "A Methodology for Utilizing Descriptive Process Models for Execution," Middle East Technical University, 2014.

[40] D. Leffingwell and D. Widrig, *Managing Software Requirements: A Unified Approach*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000.

[41] J. C. Westland, "The cost of errors in software development: evidence from industry," *J. Syst. Softw.*, vol. 62, no. 1, pp. 1–9, May 2002.

[42] M. Niazi, K. Cox, and J. Verner, "A measurement framework for assessing the maturity of requirements engineering process," *Softw. Qual. J.*, vol. 16, no. 2, pp. 213–235, 2008.

[43] O. O. Top, S. Tunalilar, and O. Demirors, "Evaluation of the Effect of Functional Similarities on Development Effort," in *Software Engineering and Advanced Applications, 2008. SEAA '08. 34th Euromicro Conference*, 2008, pp. 419–426.

[44] R. S. Kaplan and D. P. Norton, "The Balanced Scorecard-Measures that Drive Performance," *Harv. Bus. Rev.*, vol. January-Fe, pp. 71–79, 1992.

[45] R. van Solingen, V. Basili, G. Caldiera, and H. D. Rombach, "Goal Question Metric (GQM) Approach," in *Encyclopedia of Software Engineering*, John Wiley & Sons, Inc., 2002.

[46] R. L. Lynch and K. F. Cross, *Measure up! The Essential Guide to Measuring Business Performance*. London: Mandarin, 1991.

[47] M. G. Brown, *Keeping Score: Using the Right Metrics to Drive World-Class Performance*, 1st ed. Productivity Press Inc., 1996.

[48] T. H. Davenport, *Process Innovation: Reengineering Work Through Information Technology*. Harvard Business School Press, Boston, 1993.

[49] J. Becker and D. Kahn, "The Process in Focus," in *Process Management SE - 1*, J. Becker, M. Kugeler, and M. Rosemann, Eds. Springer Berlin Heidelberg, 2003, pp. 1–12.

[50] A.-W. Scheer and M. Nüttgens, "ARIS Architecture and Reference Models for Business Process Management," in *Business Process Management, Models, Techniques, and Empirical Studies*, 2000, pp. 376–389.

[51] "Gartner IT Glossary - Business process management (BPM)." [Online]. Available: http://blogs.gartner.com/it-glossary/business-process-management-bpm-2/. [Accessed: 01-May-2014].

[52] W. M. P. van der Aalst, "Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management," in *Lectures on Concurrency and Petri Nets*, 2004, vol. 3098, pp. 1–65.

[53] M. zur Muehlen, *Workflow-based Process Controlling: Foundation, Design, and Implementation of Workflow-driven Process Information Systems.*, vol. 6. Berlin: Logos, 2004.

[54] H. Stachowiak, *Allgemeine Modelltheorie [General Model Theory]*. Springer, 1973.

[55] "OMG Unified Modeling Language (OMG UML), Superstructure Version 2.2 formal/2009-02-02," 2009.

[56] A.-W. W. Scheer, *Aris-Business Process Frameworks*, 2nd ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.

[57] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[58] "Business Process Model and Notation (BPMN) formal/2011-01-03," 2011.

[59] "Web Services Business Process Execution Language Version 2.0," 2007.

[60] "Standard for Integration Definition for Function Modeling (IDEF0)," 1993.

[61] R. J. Mayer, C. P. Menzel, M. K. Painter, P. S. Witte, T. Blinn, and B. Perakath, "Information Integration For Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report. Interim Technical Report AL-TR-1995-XXXX," Texas, 1995.

[62] T. M. Friel, P.G. and Blinn, "Automated IDEF3 and IDEF4 Systems Design Specification Document," 1989.

[63] G. A. Hansen, "Tools for business-process reengineering," *Software, IEEE*, vol. 11, no. 5, pp. 131–134, 1994.

[64] I. Bider, "Choosing approach to business process modeling-practical perspective," *J. Concept. Model.*, no. 34, pp. 1–16, 2005.

[65] W. M. P. van der Aalst, M. Dumas, F. Gottschalk, A. H. M. ter Hofstede, M. La Rosa, and J. Mendling, "Preserving Correctness During Business Process Model Configuration," *Form. Asp. Comput.*, vol. 22, no. 3, pp. 459–482, 2010.

[66] J. L. Peterson, "Petri Nets," *ACM Comput. Surv.*, vol. 9, no. 3, pp. 223–252, 1977.

[67] N. Russell, A. Ter Hofstede, and N. Mulyar, "Workflow controlflow patterns: A revised view," BPM-06-22, 2006.

[68] C. Gane and T. Sarson, *Structured Systems Analysis: Tools and Techniques*. McDonnell Douglas Systems Integration Company, 1977.

[69] N. Chapin, "Flowcharting With the ANSI Standard: A Tutorial," *ACM Comput. Surv.*, vol. 2, no. 2, pp. 119–146, 1970.

[70] A. Caetano, A. R. Silva, and J. Tribolet, "Using roles and business objects to model and understand business processes," *Proc. 2005 ACM Symp. Appl. Comput. - SAC '05*, p. 1308, 2005.

[71] M. Salmanoğlu, O. Demirörs, and O. Türetken, “Exploration of a Method for COSMIC Size Estimation from S-BPM,” in *S-BPM ONE - Application Studies and Work in Progress SE - 4*, vol. 422, C. Zehbold, Ed. Springer International Publishing, 2014, pp. 55–66.

[72] R. Lu and S. Sadiq, “A survey of comparative business process modeling approaches,” in *10th International Conference, BIS 2007*, 2007, pp. 82–94.

[73] J. Mendling, J. Recker, and H. Reijers, “On the usage of labels and icons in business process modeling,” *Int. J. Inf. Syst. Model. Des.*, vol. 1, no. 2, pp. 40–58, 2010.

[74] P. Fettke, P. Loos, and J. Zwicker, “Business process reference models: Survey and classification,” *Bus. Process Manag. Work.*, pp. 469–483, 2006.

[75] M. Pesic and W. M. P. Van Der Aalst, “Towards a Reference Model for Work Distribution in Workflow Management Systems,” in *Proceedings of the Workshop on Business Process Reference Models (BPRM 2005)*, 2005.

[76] A. Tsai, J. Wang, W. Tepfenhart, and D. Rosca, “EPC Workflow Model to WIFA Model Conversion,” in *Systems, Man and Cybernetics, 2006. SMC ’06. IEEE International Conference on*, 2006, vol. 4, pp. 2758–2763.

[77] E. Kindler, “On the semantics of EPCs: Resolving the vicious circle,” *Data Knowl. Eng.*, vol. 56, no. 1, pp. 23–40, Jan. 2006.

[78] J. Mendling, W. van der Aalst, and W. Vienna, “Towards EPC Semantics based on State and Context,” in *Proceedings of the 5th EPC Workshop*, 2006.

[79] J. Mendling, G. Neumann, and M. Nüttgens, “Towards Workflow Pattern Support of Event-Driven Process Chains (EPC,” in *Proc. of the 2nd Workshop XML4BPM 2005*, 2005, pp. 23–38.

[80] W. M. P. van der Aalst, J. Desel, and E. Kindler, “On the Semantics of EPCs: A Vicious Circle,” in *Proceedings of the EPK 2002: Business Process Management using EPCs*, 2002, pp. 71–80.

[81] IDS Scheer AG., *ARIS Method. ARIS 6 Collaborative Suite 6.2.3*. 2004.

[82] S. A. White, “Introduction to BPMN.” 2006.

[83] M. zur Muehlen and J. Recker, “How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation,” in *Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE’08)*, 2008, vol. 5074, pp. 465–479.

[84] M. Weske, “Business Process Modeling and Analysis Course,” *OpenHpi*, 2013. [Online]. Available: https://openhpi.de/courses/b93ba381-c3f9-4233-8424-f66eca8d4d8b. [Accessed: 02-May-2014].

[85] C. Menzel and R. Mayer, “The IDEF Family of Languages,” in *Handbook on Architectures of Information Systems SE - 10*, P. Bernus, K. Mertins, and G. Schmidt, Eds. Springer Berlin Heidelberg, 1998, pp. 209–241.

[86] Agile Modeling, “UML 2 Activity Diagramming Guidelines.” [Online]. Available: http://www.agilemodeling.com/style/activityDiagram.htm. [Accessed: 02-May-2014].

[87] N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and P. Wohed, “On the Suitability of UML Activity Diagrams for Business Process Modelling,” in *Proceedings of the Third Asia-Pacific Conference on Conceptual Modelling (APCCM 2006)*, 2006, vol. 53, pp. 95–104.

[88] Workflow Patterns Initiative, “Workflow Patterns.” [Online]. Available: http://www.workflowpatterns.com/. [Accessed: 02-May-2014].

[89] M. Indulska, J. Recker, M. Rosemann, and P. F. Green, “Representational Deficiency of Process Modelling Languages: Measures and Implications.,” in *ECIS*, 2008, pp. 1632–1643.

[90] S. Stein, “EPC vs. BPMN - the perfect flamewar,” 2010. [Online]. Available: http://www.ariscommunity.com/users/sstein/2010-04-15-epc-vs-bpmn-perfect-flamewar. [Accessed: 02-May-2014].

[91] J. Recker, M. Indulska, M. Rosemann, and P. Green, “How Good is BPMN Really? Insights from Theory and Practice,” in *14th European Conference on Information Systems*, 2006, pp. 1–12.

[92] A. Gross and J. Doerr, “EPC vs. UML Activity Diagram - Two Experiments Examining their Usefulness for Requirements Engineering,” in *Requirements Engineering Conference, 2009. RE ’09. 17th IEEE International*, 2009, pp. 47–56.

[93] B. List and B. Korherr, “An evaluation of conceptual business process modelling languages,” in *SAC ’06 Proceedings of the 2006 ACM symposium on Applied computing*, 2006, no. Section 3, pp. 1532–1539.

[94] M. zur Muehlen and M. Indulska, “Modeling languages for business processes and business rules: A representational analysis,” *Inf. Syst.*, vol. 35, no. 4, pp. 379–390, Jun. 2010.

[95] J. Recker and A. Dreiling, “Does it matter which process modelling language we teach or use? an experimental study on understanding process modelling languages without formal education,” in *18th Australasian Conference on Information Systems*, 2007, pp. 356–366.

[96] J. Recker, “Opportunities and constraints: the current struggle with BPMN,” *Bus. Process Manag. J.*, vol. 16, no. 1, pp. 181–201, 2010.

[97] W. Tscheschner, “Transformation from EPC to BPMN,” in *Business Process Technology*, 2006.

[98] O. Levina, "Assessing Information Loss in EPC to BPMN Business Process Model Transformation," in *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2012 IEEE 16th International*, 2012, pp. 51–55.

[99] V. Kotsev, I. Stanev, and K. Grigorova, "BPMN-EPC-BPMN Converter," vol. 50, no. 6.1, pp. 54–59, 2011.

[100] ARIS Community, "Learn how to transform models." [Online]. Available: http://www.ariscommunity.com/videos/learn-how-transform-models. [Accessed: 02-May-2014].

[101] W. Wang, H. Ding, J. Dong, and C. Ren, "A Comparison of Business Process Modeling Methods," in *Service Operations and Logistics, and Informatics, 2006. SOLI '06. IEEE International Conference on*, 2006, pp. 1136–1141.

[102] H. Mili, G. Tremblay, G. B. Jaoude, É. Lefebvre, L. Elabed, and G. El Boussaidi, "Business Process Modeling Languages: Sorting Through the Alphabet Soup," *ACM Comput. Surv.*, vol. 43, no. 1, pp. 4:1–4:56, 2010.

[103] T. Andres, "From Business Process to Application: Model-Driven Development of Management Software," in *AGILITY by ARIS Business Process Management SE - 20*, A.-W. Scheer, H. Kruppke, W. Jost, and H. Kindermann, Eds. Springer Berlin Heidelberg, 2006, pp. 221–231.

[104] M. Blechar, "Magic Quadrant for Business Process Analysis Tools G00161090," 2008.

[105] S. Sadiq, M. Indulska, W. Bandara, and S. Chong, "Major Issues in Business Process Management: A Vendor Perspective," in *PACIS 2007 Proceedings*, 2007, p. 135.

[106] J. A. Zachman, "A Framework for Information Systems Architecture," *IBM Syst. J.*, vol. 26, no. 3, pp. 276–292, 1987.

[107] ESPRIT Consortium AMICE, *CIMOSA: Open System Architecture for CIM (Research Reports Esprit / Project 688/5288. AMICE)*, Extended. Springer, 1993.

[108] T. J. Williams and H. Li, "A Specification and Statement of Requirements for GERAM (The Generalised Enterprise Reference Architecture and Methodology). Report Number 159," 1995.

[109] US Department of Defence, "The DoDAF Architecture Framework Version 2.02," 2010.

[110] The Open Group, "TOGAF® Version 9.1."

[111] T. Rozman and R. V Horvat, "Case study of the suitability of the process modelling methodology in a team," in *Information Technology Interfaces, 2006. 28th International Conference on*, 2006, pp. 103–108.

[112] Y.-G. Kim, “Process Modeling For BPR: Event-Process Chain Approach,” in *ICIS 1995 Proceedings*, 1995, p. Paper 11.

[113] D. Norton, M. Blechar, and T. Jones, “What You Need to Know,” no. February 2010, pp. 1–22, 2011.

[114] BPM Academic Initiative, “Signavio Process Modeler.” [Online]. Available: http://academic.signavio.com/p/explorer. [Accessed: 03-May-2014].

[115] Visual Paradigm, “Visual Paradigm Process Modeling Tools for In-Depth Process Analysis.” [Online]. Available: http://www.visual-paradigm.com/features/business-process-modeling/. [Accessed: 03-May-2014].

[116] C. Böhme, J. Hartmann, H. Kern, and S. Kühne, “bflow* Toolbox-an Open-Source Modeling Tool,” in *Business Process Management Demonstration Track 2010*, 2010, pp. 1–6.

[117] P. Haumer, “Introducing the Eclipse Process Framework,” 2006.

[118] Bizagi, “Bizagi Process Modeler.” [Online]. Available: http://www.bizagi.com/products/bizagi-process-modeler. [Accessed: 03-May-2014].

[119] iGrafx, “iGrafx Process Modeling & Analysis.” .

[120] Sparx Systems, “Enterprise Architect Full Lifecycle Modeling.” [Online]. Available: http://www.sparxsystems.com.au/products/index.html. [Accessed: 03-May-2014].

[121] J. Pinggera, S. Zugal, and B. Weber, “Investigating the Process of Process Modeling with Cheetah Experimental Platform.,” *EMISA Forum*, vol. 30, no. 2, pp. 25–31, 2010.

[122] A. Borgida, F. Dalpiaz, J. Horkoff, and J. Mylopoulos, “Requirements models for design- and runtime: A position paper,” in *Modeling in Software Engineering (MiSE), 2013 5th International Workshop on*, 2013, pp. 62–68.

[123] C. Monsalve, A. April, and A. Abran, “On the expressiveness of business process modeling notations for software requirements elicitation,” in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, pp. 3132–3137.

[124] “Chaos—the state of the software industry,” 2003.

[125] COSMIC, “The COSMIC Functional Size Measurement Method Version 4.0 Measurement Manual,” 2014.

[126] R. F. Goldsmith, *Discovering Real Business Requirements for Software Project Success*. Norwood, MA, USA: Artech House, Inc., 2004.

[127] IEEE Computer Society, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide V3.0)*. IEEE Computer Society, 2014.

[128] IEEE, "IEEE Recommended Practice for Software Requirements Specications - IEEE Std 830-1998," New York, NY, USA, 1998.

[129] I. F. Alexander and R. Stevens, *Writing Better Requirements*. Pearson Education, 2002.

[130] H. F. Hofmann and F. Lehner, "Requirements Engineering As a Success Factor in Software Projects," *IEEE Softw.*, vol. 18, no. 4, pp. 58–66, 2001.

[131] J. Vara, M. Fortuna, J. Sánchez, C. L. Werner, and M. S. Borges, "A Requirements Engineering Approach for Data Modelling of Process-Aware Information Systems," in *Business Information Systems SE - 12*, vol. 21, W. Abramowicz, Ed. Springer Berlin Heidelberg, 2009, pp. 133–144.

[132] E. S. K. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," in *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, 1997, pp. 226–235.

[133] T. Specht, J. Drawehn, M. Thränert, and S. Kühne, "Modeling Cooperative Business Processes and Transformation to a Service Oriented Architecture," 2005.

[134] N. A. M. Maiden, S. Manning, S. Jones, and J. Greenwood, "Generating requirements from systems models using patterns: a case study," *Requir. Eng.*, vol. 10, no. 4, pp. 276–288, 2005.

[135] B. Berenbach, D. Paulish, J. Kazmeier, and A. Rudorfer, *Software & Systems Requirements Engineering: In Practice*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2009.

[136] J. Daniels and T. Bahill, "The hybrid process that combines traditional requirements and use cases," *Syst. Eng.*, vol. 7, no. 4, pp. 303–319, 2004.

[137] P. Staccini, M. Joubert, J.-F. Quaranta, D. Fieschi, and M. Fieschi, "Modelling health care processes for eliciting user requirements: a way to link a quality paradigm and clinical information system design," *Int. J. Med. Inform.*, vol. 64, no. 2–3, pp. 129–142, 2001.

[138] J. Mylopoulos, L. Chung, and E. Yu, "From Object-oriented to Goal-oriented Requirements Analysis," *Commun. ACM*, vol. 42, no. 1, pp. 31–37, 1999.

[139] D. Amyot, S. Ghanavati, J. Horkoff, G. Mussbacher, L. Peyton, and E. Yu, "Evaluating goal models within the goal-oriented requirement language," *Int. J. Intell. Syst.*, vol. 25, no. 8, pp. 841–877, 2010.

[140] D. Amyot and G. Mussbacher, "User Requirements Notation: The First Ten Years, The Next Ten Years (Invited Paper).," *JSW*, vol. 6, no. 5, pp. 747–768, 2011.

[141] M. Weiss and D. Amyot, "Business Process Modeling with URN," *Int. J. E-bus. Res.*, vol. 1, pp. 63–90, 2005.

[142] J. Lockerbie and N. A. M. Maiden, "REDEPEND: Tool Support for i* Modelling in Large-scale Industrial Projects.," in *CAiSE Forum*, 2008, vol. 344, pp. 69–72.

[143] "Objectiver." [Online]. Available: http://www.objectiver.com.

[144] V. Künzle and M. Reichert, "PHILharmonicFlows: towards a framework for object-aware process management," *J. Softw. Maint. Evol. Res. Pract.*, vol. 23, no. 4, pp. 205–244, 2011.

[145] C. M. Chiao, V. Kunzle, and M. Reichert, "Integrated modeling of process- and data-centric software systems with PHILharmonicFlows," in *Communicating Business Process and Software Models Quality, Understandability, and Maintainability (CPSM), 2013 IEEE 1st International Workshop on*, 2013, pp. 1–10.

[146] J. Vara and J. Sánchez, "Improving Requirements Analysis through Business Process Modelling: A Participative Approach," in *Business Information Systems SE - 15*, vol. 7, W. Abramowicz and D. Fensel, Eds. Springer Berlin Heidelberg, 2008, pp. 165–176.

[147] M. Jackson and P. Zave, "Deriving Specifications from Requirements: An Example," in *Proceedings of the 17th International Conference on Software Engineering*, 1995, pp. 15–24.

[148] COSMIC, "ISO/IEC 19761:2011 Software engineering -- COSMIC: a functional size measurement method," 2011.

[149] B. Marín, G. Giachetti, and O. Pastor, "Measurement of Functional Size in Conceptual Models: A Survey of Measurement Procedures Based on COSMIC," in *Software Process and Product Measurement SE - 15*, vol. 5338, R. Dumke, R. Braungarten, G. Büren, A. Abran, and J. Cuadrado-Gallego, Eds. Springer Berlin Heidelberg, 2008, pp. 170–183.

[150] K. Berg, T. Dekkers, and R. Oudshoorn, "Functional size measurement applied to UML-based user requirements," in *2nd Software Measurement European Forum, SMEF 2005*, 2005.

[151] M. Heričko, I. Rozman, and A. Živkovič, "A formal representation of functional size measurement methods," *J. Syst. Softw.*, vol. 79, no. 9, pp. 1341–1358, 2006.

[152] E. Lamma, P. Mello, and F. Riguzzi, "A System for Measuring Function Points." .

[153] M. Jenner, "COSMIC FFP 2.0 and UML: Estim ation of the size of a system specified in UML – Problems of granularity.," in *roceedings of FESMA-DESMA Conference 2001,*, 2001.

[154] G. Levesque, V. Bevo, and D. T. Cao, "Estimating Software Size with UML Models," in *Proceedings of the 2008 C3S2E Conference*, 2008, pp. 81–87.

[155] N. Condori-Fernández, S. Abrahão, and O. Pastor, "On the Estimation of the Functional Size of Software from Requirements Specifications," *J. Comput. Sci. Technol.*, vol. 22, no. 3, pp. 358–370, 2007.

[156] S. Azzouz and A. Abran, "A proposed measurement role in the Rational Unified Process (RUP) and its implementation with," in *ISO 19761: COSMIC-FFP', proceedings of Software Measurement European Forum - SMEF 2004*, 2004.

[157] O. Demirörs and Ç. Gencel, "A Comparison of Size Estimation Techniques Applied Early in the Life Cycle," *Software Process Improvement*. Springer LNCS, pp. 184–194, 2004.

[158] A. Abran, A. Lesterhuis, M. O'Neill, L. Santillo, C. Symons, and H. Toivonen, "The COSMIC Functional Size Measurement Method Version 3.0 Guideline for Sizing Business Application Software," 2008.

[159] "Information Technology Infrastructure Library (ITIL)," 2011.

[160] E. Aydin, "An Automated Tool for Quality Manual Generation from Business Process Models," Middle East Technical University, 2010.

[161] H. Dalianis, "A method for validating a conceptual model by natural language discourse generation," in *Advanced Information Systems Engineering SE - 25*, vol. 593, P. Loucopoulos, Ed. Springer Berlin Heidelberg, 1992, pp. 425–444.

[162] C. Rolland and C. Proix, "A natural language approach for Requirements Engineering," in *Advanced Information Systems Engineering SE - 15*, vol. 593, P. Loucopoulos, Ed. Springer Berlin Heidelberg, 1992, pp. 257–277.

[163] F. Meziane, N. Athanasakis, and S. Ananiadou, "Generating Natural Language specifications from UML class diagrams," *Requir. Eng.*, vol. 13, no. 1, pp. 1–18, 2008.

[164] H. Leopold, J. Mendling, and A. Polyvyanyy, "Generating Natural Language Texts from Business Process Models," in *Advanced Information Systems Engineering SE - 5*, vol. 7328, J. Ralyté, X. Franch, S. Brinkkemper, and S. Wrycza, Eds. Springer Berlin Heidelberg, 2012, pp. 64–79.

[165] M. Netjes, H. A. Reijers, and W. M. P. Van Der Aalst, "The PrICE Tool Kit : Tool Support for Process Improvement," in *Proceedings of the Business Process Management 2010 Demonstration Track*, 2010.

[166] A. Kronz, "Managing of Process Key Performance Indicators as Part of the ARIS Methodology," in *Corporate Performance Management SE - 3*, Springer Berlin Heidelberg, 2006, pp. 31–44.

[167] A.-W. W. Scheer, "Professor Scheer's Advanced BPM Assessment," 2007.

[168] D. Thiault, *Managing Performance Through Business Processes: From BPM to the practice of process management*. CreateSpace Independent Publishing Platform, 2012.

[169] P. Taticchi, *Business Performance Measurement and Management New Contexts, Themes and Challenges*. Springer Berlin Heidelberg, 2010.

[170] A. Del-Rio-Ortega and M. Resinas, "Towards Modelling and Tracing Key Performance Indicators in Business Processes," *Actas los Talleres las Jornadas Ing. del Softw. y Bases Datos*, vol. 3, no. 3, pp. 57–67, 2009.

[171] A. Pourshahid, P. Chen, D. Amyot, M. Weiss, and A. Forster, "Business Process Monitoring and Alignment: An Approach Based on the User Requirements Notation and Business Intelligence Tools," *10th Work. Requir. Eng.*, pp. 80–91, 2007.

[172] J. Mendling and M. Strembeck, "Influence factors of understanding business process models," *11th Int. Conf. BIS 2008*, vol. 7, pp. 142–153, 2008.

[173] R. Davis and E. Brabander, *ARIS Design Platform Getting Started with BPM*. Springer London, 2007, p. 370.

[174] ARIS Community, "How to use Process Interface Properly?," 2009. [Online]. Available: http://www.ariscommunity.com/users/weerakhan/2009-10-15-how-use-process-interface-properly. [Accessed: 06-May-2014].

[175] J. Mendling, G. Neumann, and W. Aalst, "Understanding the Occurrence of Errors in Process Models Based on Metrics," in *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS SE - 9*, vol. 4803, R. Meersman and Z. Tari, Eds. Springer Berlin Heidelberg, 2007, pp. 113–130.

[176] A.-W. W. Scheer, *Business Process Engineering*, 1st ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.

[177] C. Seel and D. Vanderhaeghen, "Meta-model based extensions of the epc for inter-organisational process modelling," in *IN: PROCEEDINGS 4TH GI-WORKSHOP EPK 2005 - GESCHAFTSPROZESSMANAGEMENT*, 2005, pp. 117–136.

[178] F. Lautenbacher, "Comparison of Business Process Metamodels."

[179] S. Kühne, H. Kern, V. Gruhn, and R. Laue, "Business process modeling with continuous validation," *J. Softw. Maint. Evol. Res. Pract.*, vol. 22, no. 6–7, pp. 547–566, 2010.

[180] C. Fillies and F. Weichhardt, "On Ontology-based Event-driven Process Chains," 2005.

[181] F. Yeşildoruk, B. Aysolmaz, and O. Demirörs, "The Tool Coverage of Software Process Improvement Frameworks for Small and Medium Sized Enterprises," in *Product-Focused Software Process Improvement SE - 22*, vol. 32, F. Bomarius, M.

Oivo, P. Jaring, and P. Abrahamsson, Eds. Springer Berlin Heidelberg, 2009, pp. 290–302.

[182] M. Schrepfer, "Modeling Guidelines for Business Process Models," Humboldt-Universitat zu Berlin, 2010.

[183] J. Stirna, A. Persson, and K. Sandkuhl, "Participative enterprise modeling: experiences and recommendations," in *19th International Conference, CAiSE 2007*, 2007, pp. 546–560.

[184] P. J. M. Frederiks and T. P. van der Weide, "Information modeling: The process and the required competencies of its participants," *Data Knowl. Eng.*, vol. 58, no. 1, pp. 4–20, 2006.

[185] M. La Rosa, P. Wohed, J. Mendling, A. H. M. ter Hofstede, H. a. Reijers, and W. M. P. van der Aalst, "Managing Process Model Complexity Via Abstract Syntax Modifications," *IEEE Trans. Ind. Informatics*, vol. 7, no. 4, pp. 614–629, Nov. 2011.

[186] S. Goedertier and J. Vanthienen, "Declarative Process Modeling with Business Vocabulary and Business Rules," in *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops SE - 83*, vol. 4805, R. Meersman, Z. Tari, and P. Herrero, Eds. Springer Berlin Heidelberg, 2007, pp. 603–612.

[187] M. Milanovic, D. Gasevic, and L. Rocha, "Modeling Flexible Business Processes with Business Rule Patterns," in *Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International*, 2011, pp. 65–74.

[188] M. La Rosa, A. H. M. Ter Hofstede, P. Wohed, H. A. Reijers, J. Mendling, and W. M. P. van der Aalst, "Managing Process Model Complexity via Concrete Syntax Modifications," *IEEE Trans. Ind. Informatics*, vol. 7, no. 2, pp. 255–265, 2011.

[189] Signavio, "BPMN Modeling Guidelines," 2014. [Online]. Available: http://www.modeling-guidelines.org/. [Accessed: 07-May-2014].

[190] D. Cohn and R. Hull, "Business Artifacts: A Data-centric Approach to Modeling Business Operations and Processes," *IEEE Data Eng. Bull.*, vol. 32, no. 3, pp. 3–9, 2009.

[191] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1997.

[192] C. Monsalve, A. Abran, and A. April, "Functional Size Measurement with Business Process Models: the Business Application Domain," in *Joined International Conferences on Software Measurement – IWSM/MetriKon/Mensura 2010, Stuttgart, Germany, November 10-12.*, 2010, pp. 277–290.

[193] B. Aysolmaz and O. Demirörs, "A Detailed Software Process Improvement Methodology: BG-SPI," in *Systems, Software and Service Process Improvement SE - 9*, vol. 172, R. O'Connor, J. Pries-Heje, and R. Messnarz, Eds. Springer Berlin Heidelberg, 2011, pp. 97–108.

[194] Informatica, "Business Glossary: Create, Manage, and Share Business Meaning and Context," 2012. [Online]. Available: http://www.informatica.com/Images/06922_business-glossary_ds_en-US.pdf. [Accessed: 12-May-2014].

[195] M. Modell, *A Professional's Guide to Systems Analysis, Second Edition*. New York, NY, USA: McGraw-Hill, 1996.

[196] T. Stahl, M. Voelter, and K. Czarnecki, *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons, 2006.

[197] The Eclipse Foundation, "Eclipse Modeling Framework Project (EMF)," 2014. [Online]. Available: http://www.eclipse.org/modeling/emf/. [Accessed: 12-May-2014].

[198] The Eclipse Foundation, "Graphical Modeling Project (GMP)," 2014. [Online]. Available: http://www.eclipse.org/modeling/gmp/. [Accessed: 12-May-2014].

[199] iText Software Corp., "ITEXT PDF Library," 2014. [Online]. Available: http://itextpdf.com/. [Accessed: 11-May-2014].

[200] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, Inc, 2008, p. 296.

[201] R. K. Yin, *Case Study Research: Design and Methods, 3rd Edition (Applied Social Research Methods, Vol. 5)*. SAGE Publications, Inc, 2002, p. 200.

[202] I. Benbasat, D. K. Goldstein, and M. Mead, "The Case Research Strategy in Studies of Information Systems," *MIS Q.*, vol. 11, no. 3, pp. 369–386, Sep. 1987.

[203] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, 2009.

[204] "SMRG, Software Management Research Group, Informatics Institute, METU." [Online]. Available: http://smrg.ii.metu.edu.tr/smrgp/. [Accessed: 23-Mar-2014].

[205] E. Ungan, O. Demirörs, Ö. Top, and B. Özkan, "An Experimental Study on the Reliability of COSMIC Measurement Results," in *Software Process and Product Measurement SE - 23*, vol. 5891, A. Abran, R. Braungarten, R. Dumke, J. Cuadrado-Gallego, and J. Brunekreef, Eds. Springer Berlin Heidelberg, 2009, pp. 321–336.

[206] O. O. Top, O. Demirors, and B. Ozkan, "Reliability of COSMIC Functional Size Measurement Results: A Multiple Case Study on Industry Cases," in *Software Engineering and Advanced Applications, 2009. SEAA '09. 35th Euromicro Conference on*, 2009, pp. 327–334.

[207] B. Aysolmaz, "Unified Business Process Modeling Methodology (UPROM) Application: Case Study Results METU/II-TR-2014-32," Ankara, Turkey, 2014.

[208] O. Turetken, O. Ozcan Top, B. Ozkan, and O. Demirors, “The Impact of Individual Assumptions on Functional Size Measurement,” in *Software Process and Product Measurement SE - 14*, vol. 5338, R. Dumke, R. Braungarten, G. Büren, A. Abran, and J. Cuadrado-Gallego, Eds. Springer Berlin Heidelberg, 2008, pp. 155–169.

[209] C. Symons, “The Performance of Real-Time , Business Application and Component Software Projects,” 2011.

[210] Ç. Gencel and C. Symons, “How Can We Improve Project Effort Prediction from FSM Data?,” in *MENSURA*, 2013.

[211] C. Jones, *Estimating Software Costs: Bringing Realism to Estimating*, 2nd ed. McGraw-Hill Osborne Media, 2007.

[212] B. Aysolmaz, D. İren, and O. Demirörs, “An Effort Prediction Model Based on BPM Measures for Process Automation,” in *Enterprise, Business-Process and Information Systems Modeling SE - 12*, vol. 147, S. Nurcan, H. Proper, P. Soffer, J. Krogstie, R. Schmidt, T. Halpin, and I. Bider, Eds. Springer Berlin Heidelberg, 2013, pp. 154–167.

CURRICULUM VITAE

PERSONAL INFORMATION

Banu Aysolmaz was born in İzmir, Turkey in 1981 She received her bachelor degree from Industrial Engineering in Middle East Technical University in 2003. She received her M.S. degree from Information Systems in Informatics Institute of Middle East Technical University in 2007. She has been working as an industrial engineer since 2003. Her research interests include process improvement, business process modeling and software engineering. You can contact her at banuays@gmail.com.

WORK EXPERIENCE

2009 – Present, Bilgi Grubu Ltd. Software process improvement and business process modeling consultant, trainer and project manager in various organizations and projects.

2009 – 2013. METU Informatics Institute. Research assistant.

2007 – 2008. Çukurova Development Agency. Expert in project department.

2005 – 2006. SDT Space and Defense Technologies. Project manager and group leader in Simulation Software Department

2003 – 2005. SATEK Defense Technologies. Software engineer and quality manager.

PUBLICATIONS

B. Aysolmaz and O. Demirörs, "Automated Functional Size Estimation using Business Process Models with UPROM Method," in Software Process and Product Measurement, 2014. In press.

B. Aysolmaz and O. Demirörs, "UPROM Tool : A Unified Business Process Modeling Tool for Generating Software Life Cycle Artifacts," in Proceedings of the CAiSE Forum 2014, 2014. In press.

B. Aysolmaz and O. Demirörs, "Modeling Business Processes to Generate Artifacts for Software Development : A Methodology," in Modeling in Software Engineering (MISE), 2014 ICSE Workshop on, 2014. DOI: 10.1145/2593770.2593775.

B. Aysolmaz, “Unified Business Process Modeling Methodology (UPROM) Application: Case Study Results METU/II-TR-2014-32,” Ankara, Turkey, 2014. http://expertjudgment.com/publications/METU_II_TR_2014_32.pdf

B. Aysolmaz, D. İren, and O. Demirörs, “An Effort Prediction Model Based on BPM Measures for Process Automation,” in Enterprise, Business-Process and Information Systems Modeling SE - 12, vol. 147, S. Nurcan, H. Proper, P. Soffer, J. Krogstie, R. Schmidt, T. Halpin, and I. Bider, Eds. Springer Berlin Heidelberg, 2013, pp. 154–167.

B. Aysolmaz and O. Demirörs, “A Detailed Software Process Improvement Methodology: BG-SPI,” in Systems, Software and Service Process Improvement SE - 9, vol. 172, R. O‘Connor, J. Pries-Heje, and R. Messnarz, Eds. Springer Berlin, 2011, pp. 97–108.

B. Aysolmaz, A. Coskuncay, O. Demirörs, and A. Yildiz, “Kamuda İş Süreçleri Modelleme: Gereği ve Yararları,” in 5. Ulusal Yazılım Mühendisliği Sempozyumu - UYMS’11, 2011.

B. Aysolmaz, A. Yildiz, and O. Demirors, “BG-SPI: Yinelemeli Yazılım Süreç İyileştirme Yöntemi,” in 5. Ulusal Yazılım Mühendisliği Sempozyumu - UYMS’11, 2011.

A. Coskuncay, B. Aysolmaz, O. Demirors, O. Bilen, and I. Dogan, “Bridging The Gap Between Business Process Modeling And Software Requirements Analysis: A Case Study,” MCIS 2010 Proc., p. Paper 20, 2010.

A. Coskuncay, B. Aysolmaz, O. Demirörs, O. Bilen, and I. Dogan, “Eşzamanlı İş Süreci Modelleme ve Gereksinim Analizi için Bir Yaklaşım,” in 2. Yazılım Kalitesi ve Yazılım Geliştirme Araçları Sempozyumu, 2010.

F. Yeşildoruk, B. Aysolmaz, and O. Demirors, “Küçük ve Orta Ölçekli İşletmeler için Geliştirilen Süreç İyileştirme Çerçevelerinin Kapsamı,” in Ulusal Yazılım Mühendisliği Sempozyumu, 2010.

F. Yeşildoruk, B. Aysolmaz, and O. Demirörs, “The Tool Coverage of Software Process Improvement Frameworks for Small and Medium Sized Enterprises,” in Product-Focused Software Process Improvement SE - 22, vol. 32, F. Bomarius, M. Oivo, P. Jaring, and P. Abrahamsson, Eds. Springer Berlin Heidelberg, 2009, pp. 290–302.

B. Aysolmaz and O. Demirors, “Simülasyon Sistemlerinde Kavramsal Modellerin Tasarımda Kullanılması – Bir Vaka Çalışması,” in 4. SAVTEK Savunma Teknolojileri Kongresi, 2008.

B. Aysolmaz, “Conceptual Model of a Synthetic Environment Simulation System Developed Using Extended KAMA Methodology METU/II-TR-2007-17,” Ankara, Turkey, 2007.

CERTIFICATES

2010 – COSMIC Software Functional Size Measurer; COSMIC

2011 – Introduction to CMMI for Development v1.3; SEI

2011 – PMI-Agile Certified Professional (PMI-Agile); PMI

2012 – Standard CMMI Appraisal Method for Process Improvement Training v1.3; SEI

2013 – Business Process Modeling and Analysis; University of Potsdam

INTERESTS

Traveling (member of Nature Travelers Association), outdoor sports, adventure racing (member of team Rock Lizards), long distance running (Runtalya Marathon 2014 finisher), orienteering, mountain biking (2014 Turkish elite women MTB orienteering champion), mountaineering, rock climbing, yoga, table tennis (METU team member), archeology, photography, music (classic guitar player).