A METHODOLOGY FOR UTILIZING DESCRIPTIVE PROCESS MODELS FOR EXECUTION

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

BY

MAKBULE FİLİZ ÇELİK

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

MARCH 2014

Approval of the thesis:

A METHODOLOGY FOR UTILIZING DESCRIPTIVE PROCESS MODELS FOR EXECUTION

submitted by **MAKBULE FILIZ ÇELIK** in partial fulfillment of the requirements for the degree of **Philosophy of Doctorate in Information Systems Department, 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, **Department of Information Systems** ________________

Examining Committee Members:

Prof. Dr. Semih Bilgen
EEE, METU ________________

Prof. Dr. Onur Demirors
IS, METU ________________

Prof. Dr. Ali H. Doğru
CENG, METU ________________

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

Assist. Prof. Dr. O. Ozgur Tanrıover
CENG, AU ________________

Date: **06.03.2014**

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: MAKBULE FİLİZ ÇELİK

Signature :

ABSTRACT

A METHODOLOGY FOR UTILIZING DESCRIPTIVE PROCESS MODELS FOR EXECUTION

Çelik, Makbule Filiz
Ph.D., Department of Information Systems
Supervisor : Prof. Dr. Onur Demirörs

March 2014, 127 pages

Process models are in the center of unifying various views enhancing organizational efficiency. In its current state, different models are produced for different purposes for the same processes. This diversity results in redundancy for modeling efforts and inconsistencies among various models. This study proposes a methodology for organizations to perform business process modeling in a way that produced models can be used both for process description and process execution. Instead of producing different models for process definition and process execution, our methodology focuses on the idea that a single process model can be utilized for both purposes. Each modeling purpose has its own perspective which enables modelers to focus on relevant aspects of the model. Our method guides the modelers with the order of modeling activities as well as the produced artifacts. A multiple case-study is conducted in order to evaluate the application of the method for utilizing process models for execution and to validate the results.

Keywords: process modeling, process execution, business process management

ÖZ

TANIMLAYICI SÜREÇ MODELLERİNİN OTOMASYON İÇİN KULLANILMASINI SAĞLAYAN METODOLOJİ

Çelik, Makbule Filiz
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi : Prof. Dr. Onur Demirörs

Mart 2014, 127 sayfa

Bu çalışma, organizasyonların sureç modelleme çalışmaları sonucunda uretilen modellerin sureç otomasyonu amacıyla kullanılabilmelerini sağlayacak bir metot onermektedir. Metot, farklı amaçlar için farklı modeller uretmek yerine, tek bir modelin her iki amaç için de kullanılabileceği fikrine dayanmaktadır. Modelleme amacına uygun olarak modelleme perspektiflerinin oluşturulması ile her kullanıcı ilgili olduğu ozelliklerden sorumlu olacaktır. Metot, kullanıcılara modelleme faaliyetleri ve uretilecek urunlerle ilgili yol gosterici rol oynamaktadır. Bu çalışma, metodun uygulanabilirliğinin gosterimi ve beklenen yararların gerçekleştiğinin doğrulanması amacıyla çoklu-ornek olay incelemesi içermektedir.

Anahtar Kelimeler: sureç modelleme, sureç otomasyonu, iş sureçleri yonetimi

to my family

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor Onur Demirors for his encouragement, patience and insight throughout this thesis research.

I would also like to thank my committee members Semih Bilgen and Ali Doğru for their supportive comments and suggestions over the years. Banu Aysolmaz provided me with great support for the case studies, I am grateful for her sincere concern.

I am extremely grateful to my family for their continuous support during my studies.

And finally, I would like to express my deepest gratitude to Gunay for his never-ending support and patience.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In today's competitive circumstances, it is critical for organizations to have defined processes in order to successfully compete with its rivals. According to Gartner, the number one priority for CIOs is to build up business process capabilities within their organizations [1]. Organizations utilize business process modeling in order to define, document and communicate their processes. Regardless of the nature of these processes, these models should be as effective and efficient as possible. Since considerable amount of effort is spent on the creation of process models, organizations want to use them for many purposes ranging from simply documenting existing ones to re-designing existing processes to be able to execute them [2][3][4][5]. Since the models are intended to be used by different stakeholders with different perspectives in organizations, they should be understandable by different interest groups. The variety of intended users and focus of process models makes process modeling a complex task, it also introduces many challenges for the management of created process models.

In their study, Ould proposes that a business process model can be used to achieve three main goals, describing the processes, analyzing them and using them for process enactment [2]. Defining and analyzing the processes are concepts mainly used in business process improvement (BPI) initiations. Process enactment or executions is in the focus of Business Process Management (BPM). Business Process Management (BPM) is a discipline which helps organizations to manage their processes with a continuous improvement perspective. Studies show that BPM enhances process improvement initiations [6]. Therefore, it is clear that BPM and BPI are not separate domains but rather interrelated disciplines in the context of improving organizations' processes. However, although the core objective of these two domains is similar, improving the efficiency of the organization by improving its processes, the paths taken by them diverge considerably. The only common point utilized by both is the process modeling itself.

The common case for organizations is keeping several co-existing models of the processes for different purposes such as BPI and BPM. This separation results in several other challenges such as keeping the consistency of different models or ensuring these models are accurate and up-to-date. In this research, we propose a method for process modeling with different abstraction levels for different user groups with BPI and BPM perspectives. This thesis demonstrates how business processes may be modeled by different users and yet contain the relevant and necassary information for different perspectives.

1.1 The Context

Organizations are increasingly using process modeling to understand the ”as-is” processes and navigate through the ”to-be” processes. One of the major goals of BPM is ensuring that this navigations is carried in an effective and efficient manner. Business Process Management has become a top business priority for systems interoperability and execution of end-to-end complex processes [7][8][9]. Aalst et al. describes BPM as ”supporting business processes using methods, techniques and software to design, enact, control and analyze operational processes involving humans, organizations, applications, documents and other sources of information” [10]. A typical BPM lifecycle includes phases for business process modeling, process execution, process monitoring and process improving. Business process execution is often embedded within BPM solutions and uses business process models that are captured, analyzed and optimized by business people. Apart from the advantages of BPM, business process execution, in itself, offers strong competitive advantage for organizations by simply automating manual processes. The terms process automation and process execution are used interchangeably in the context of business process management, however they serve different purposes. The term process automation, in the context of this study, refers to the whole process in which a model is produced and then is deployed through a process engine, so that it will be executed on a real-time basis. Business process automation requires orchestration in the organization to centralize its enterprise architecture, as well as integration with many systems in the organization. It also includes process execution. The term process execution is used as the work done so that the modeled processes will be able to be deployed to a process engine. In other words, process execution is a step in process automation.

Another popular concept in creating effective business processes is Business Process Improvement (BPI) which offers a systematic approach to help an organization make significant changes in the way it does business. In most cases, organizations who are interested in BPM have two major interests; continuously improving the current processes using information technology and documenting the processes. It is very clear that BPM and BPI serve the same purpose, albeit in different approaches. The key difference between BPM and BPI life-cycle is the execution step. Execution may enhance process improvement by providing KPI measurements as inputs; however, most organizations believe that process execution is a costly step, therefore avoid the benefits.

As Figure 1.1 depicts, the first and most important step in both BPM and BPI life-cycles is defining and modeling the processes [10]. A process may be defined as a structure which consists of logically connected tasks, operators and information [10]. Basically, it has a starting event with inputs and resulting event with the outputs. Business process models are useful for understanding, learning and teaching, monitoring, measuring, improving and executing the as-is processes of an organization. Process models represent the aspects in a process, which

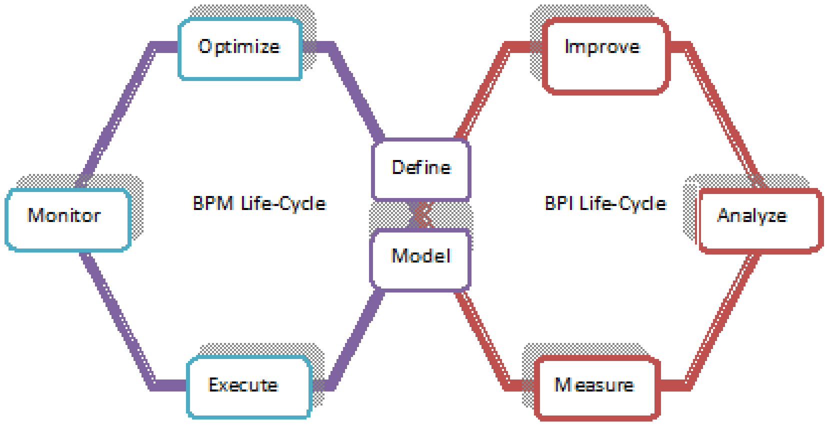


Figure 1.1: BPM and BPI life-cycles

are considered important to the purpose of the modeling. When this purpose has a wide range, the notation to emphasize the important aspects of the model will require a wide spectrum of methods, ranging from informal i.e. natural language to formal i.e. computer programs.

Process models are in the center of unifying various views enhancing organizational efficiency. Business users focus on understanding the process and the modeled elements reveal the process together with its organizational concern. They may be subject to interpretation in terms of judgment or expertise of the modelers. However, in process execution, process modeling requires to be specific and exact, focusing more on sequences, data definitions, input/output relations and integration with organizationâs technological infrastructure. Therefore, the produced models are different in semantics and structure. Although business users have domain-specific skills, they have difficulty in expressing the operational details of the processes so that the models can be executed by engines. On the other hand, IT professionals do not have domain knowledge therefore they cannot model the processes in a high level which would provide a common understanding between stakeholders. Currently, using BPMS and modeling executable processes requires a high level of expertise in business and IT, rendering existing process modeling languages unsuitable for the business user. Since it is not realistic to assume that both business and IT professionals will be equipped with the concepts of both domains, the very basic solution is to give the responsibility of different modeling steps to relevant process owners. Although the solution seems straightforward, the implications are quite complex. Hence there's a need for a business process modeling solution that both business and IT users can use in a way which makes the user interaction and means easy to understand in the context of the modeling language and easy to deploy, implement, and execute processes in a tooling context. Business users must be enabled to express activity requirements rather than to specify services and kept free from execution details.

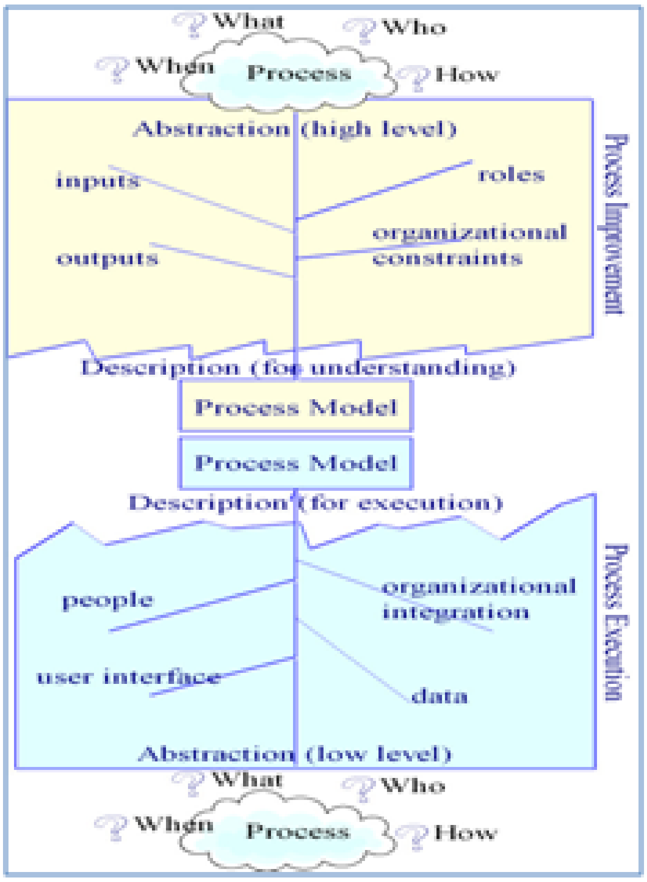


Figure 1.2: Process Models in BPM and BPI

1.2 The problem

Although BPM is very popular among organizations, it is a very challenging task to accomplish. According to a Gartner report, only a very small proportion of organizations involved with BPM initiations will be successful for the execution of processes in the near future [11].The major challenge in utilizing process models for various uses is the fact that the aim of these models and the user groups are different. When this is the case, the common practice for organizations is keeping several co-existing models of the same process for different purposes [12]. However, this results in various challenges for maintaining the consistency among the models.

When many stakeholders have interest in the same process, it is crucial that these processes must be defined in a very flexible way and there should be a communication path to ensure no misunderstandings will occur between humans and machines. The communication is very important, because in today's competitive world, self-empowerment is very important for highly skilled people in terms of their motivation. Also, Armour [13] states that many problems may arise when processes are defined by people who do not actually use it. Instead of a central control of processes, each individual wants to do his work separately in the way he knows best. This division of work, however, requires a mechanism to work on the same process together. This collaboration is not maintained by any structure in current organizations.

Business Process Management Systems (BPMS) focus on the business processes but intended users are mainly IT people, since using them requires extensive technical knowledge. As our previous study shows, adapting to BPMS is very difficult because of lack of standards [14]. It is revealed that the main users for BPM suites are those with technical knowledge since it is very difficult for business experts to understand the frameworks. However, the owners of the processes are business people; therefore, the BPM suites must be incorporated into a more business-logic perspective so that the owners can have more control over their processes.

In this study, we spotlight the problem of defining a single process model which can be used for improvement and process execution. We aim to produce inputs for business process management and execution frameworks, but our focus is on producing process models which will still be able to be used by business users.

Our solution provides a methodology for modeling the processes with a two-level perspective. Domain experts are directly involved in the modeling phase producing high-level descriptions for increased understandability and expressiveness. IT people will produce a detailed process model with semantics and execution details. The process model itself will handle the communication paradigms and therefore, each participant will be able to evaluate and modify the process. In this manner, a decentralized and self-maintainable process structure will be settled in the organization.

1.3 The solution: The PM4E Method

When the process modeling initiatives in most organizations are concerned, the process usually proceeds as follows. Firstly, IT and business users gather and they try to understand the nature of the problem and its boundaries. After initial meetings, the processes are modeled, generally by IT people, since the business users are not skilled enough to use many modeling tools. They are somehow intimidated by various notations and they hesitate to define their processes by modeling and instead they stick to traditional method of explaining the process by telling it. Business users become more involved after they see the modeled as-is processes. In the next step, business and IT users elaborate on the process, identifying missing points and eliminating ambiguities. After this, IT people understand what they are supposed to do and focus on executing other phases of software processes. On the other hand, business users, with the relaxation of being able to tell what they want, return to their main tasks. In both cases, the process models are left aside and forgotten until a maintenance issue arises in which case they become obsolete. The main reason behind this waste is the perception of the business models. Business users perceive the models as an input for IT people rather than explicit definitions of their own tasks. IT people, on the other hand, perceive the models as an understanding media in order to be able to develop the required software. However, when we consider the processes as the sole and very basic commodity of the organizations, it is obvious that they belong to everybody in the organization. In this study we propose a methodology to generate this understanding, that the processes belong to both business and IT users, and provide means for its sustainability.

As mentioned before, employees are most productive when they do what they know best. The way to let employees work separately but at the same time have a complete and correct process is summarized as follows:

- With an initial meeting consisting business and IT experts, define roles, inputs/outputs, goals in the process i.e. time constraints.
- Business users will model the process with explicit roles, major activities and input/outputs.
- Meeting with business and IT experts to reveal relations with other organizational systems, properties of input/outputs, timing constraints, error paths.
- IT experts will extend the model to include execution variables.
- Model will be directed to execution engines.
- According to execution results, the process will be improved.

The major challenge emerged by the above steps is enabling users to use the modeling tools and notations according to their habits or preferences. When current practices of BPM are inspected, it is very clear that organizations should take the whole package offered by the vendor. These packages include their own modeling and execution environments. Organizations work with the vendors to make the necessary customizations. Most of the BPM suites require modeling the processes in their own environment according to their propriety approaches. These models are therefore not usable for other initiatives in the organizations. When we analyze open source solutions for BPM, as discussed in a previous study, we see that the situation is not very different [14]. Our solution to this problem is modularizing the modeling processes. We define the methodology for modeling both for business users and IT professionals, however the modeling tool or notations is left to user preference. Because, forcing people to go in one direction, when they actually have another direction in mind does not yield expected results. Therefore, we give the modelers the freedom of choosing their environment or notation. It is certain that flexibility of this choice bring extra work in unifying them in order to work in harmony. We propose adapter mechanisms for unification.

In order to be able use solutions produced for different problems and targeting people from different expertise areas in a maintainable way, we propose a lego-like structure as a solution, as depicted in 1.3. In such a structure, each user group can externalize their expertise. It is only possible in such a structure that different user groups can accept the processes and models as their own and work willingly and efficiently. However, it should be kept in mind that not all modeling notations can be interchanged between each other. Therefore, we focus on using adapter mechanism only for control-flow based modeling notations, more specifically eEPC and BPMN. Business users may model their processes with different notations such as eEPC and BPMN. PM4E provides guidelines and a methodology for these users regardless of these notations. However, in order to be able to utilize these models for execution, these models need to be serialized to be used for enhancements regarding execution parameters. Similarly, processes may be executed using various process engines such as Apache or Activiti. Although PM4E does not require a specific engine, it is necassary to build the interfaces to transform the models into relevant executable format understandable by the process engine.

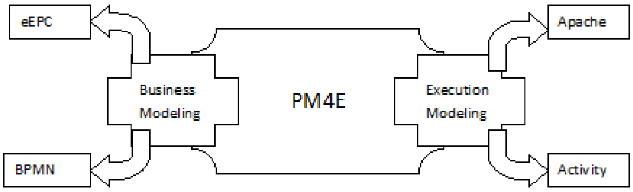


Figure 1.3: PM4E Overview

The objective of this research is to develop a methodology for organizations, which will involve in business process improvement and business process management initiations, to enable process modeling in a way that will be used by both applications. In our first exploratory case study, we have inspected and evaluated current practices and basing on the findings, developed a method. In order to evaluate the applicability of the method and validate the expected results, we have designed a multiple-case study involving two case studies. The case studies were conducted in a governmental organization. In this study, we have analyzed the shortcomings of the method and improved it accordingly. In the second case, we have evaluated the applicability and the benefits of the methodology.

1.4 Research Strategy

The research objective of this study is to develop a method for organizations involving in business process management initiations to enable performing process modeling from the description phase to the execution phase without losing the understandability of the process. In order to answer our research questions, initially we have applied a theory-based approach. We have evaluated current modeling techniques and basing on the findings we have developed a new method.

The methodology of this thesis may be split into four main parts.

1. Literature review: Related research on various types and purposes business process

modeling is searched initially. It is done using keyword search on literature search engines and extending the search by retrieving citations in the current research.

2. Establishing the method: Basing on the literature review, we have identified the lacking points and improvement opportunities regarding process modeling for execution. Focusing on these issues, we have developed a methodology to overcome the mentioned difficulties.

3. Evaluating the method: In order to verify the applicability of the method, and validate the expected benefits, a multiple-case study involving two case studies was conducted.

4. Discussion: The results of the case studies are discussed in regard with the related research questions.

1.5 Road Map

The remainder of this thesis proceeds as follows.

The following chapter introduces related research in the areas of process modeling as well as business process management and business process improvement. In this chapter, we will present the theoretical background that is necessary for accomplishing this study. The advantages and lacking issues of the related works will be analyzed and described.

Chapter 3 mentions the focus of this thesis and describes the proposed solution in detail. The phases, its activities, the roles of the participants will be discussed.

Chapter 4 will present implementation using three case studies. Not only the implementations but the lessons learned and discussions will also be presented in this chapter.

Chapter 5 will discuss the overall achievements and conclude with future directions.

CHAPTER 2

RELATED RESEARCH

The term business process has been the common concept in many fields such as, business process management (BPM), business process reengineering (BPR), business process improvement (BPI), workflow management (WfM), enterprise modeling (EM), and process innovation. Davenport [15] defines a (business) process as a structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus's emphasis on what. A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for action. Taking a process approach implies adopting the customer's point of view. Processes are the structure by which an organization does what is necessary to produce value for its customers. Davenport's definition views clear boundaries to the business processes where it has clear inputs and outputs; consists of activities, which are ordered in time and space; and should provide a value to the customer. Hammer and Champy [16] summarizes the definition of the business process as a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer. Another precise view to the processes defines it as a set of partially ordered steps or activities intended to reach a common goal [17] [18]. Laudon et al. [19] points out the peculiarity of business processes and view processes as the unique ways in which organizations coordinate and organize work activities, information, and knowledge to produce a valuable product or service. Finally Johansson et al. [19] define a process as a set of linked activities that take an input and transform it to create an output. Ideally, the transformation that occurs in the process should add value to the input and create an output that is more useful and effective to the recipient either upstream or downstream.

Combining all these definitions we may summarize that a business process must:

- be definable with clear boundaries, inputs and outputs
- consist of activities which may be ordered
- have a recipient which will benefit from the outcome
- have a place in the organizational structure

In this study, we focus on business process modeling with the focus of utilizing benefits of process improvement and process management. With this focus, this chapter summarizes the literature related to business process modeling, business process improvement and business process management. First section gives information about business process improvement. Second section describes the main elements in business process management. Since process modeling is the major concern of this study, third section gives detailed information about different tools, notations and purposes of business process modeling detailing its relations with business process improvement and management.

2.1 Business Process Improvement

Business Process Improvement (BPI) can be described as a set of organized approaches and related tools that guide organizations to enhance their performance [20]. The objective of having better results with lower costs has always been a desirable outcome for the organizations, starting from late 18th century. However, over the past hundred years, the concept of process improvement has been a hot topic and especially after 1980s, the terms efficiency, effectiveness and quality led to the development of many formal process improvement methodologies and standards such as Six Sigma, Total Quality Management, ISO 9000 and Business Process Reengineering. Process improvement starts with making the processes explicit in order to understand them and increase communication.

Process improvement may be interpreted in two ways, first improving the process itself and second improving what the process produces [21]. Regardless of the focus of improvement - the process or the product - the improvement begins with analyzing the process and defining it in order to identify improvement opportunities. Indulska et al shows that the top three benefits expected from process modeling are process improvement, understanding and communication [22]. These concepts are all related to continuously improving an organizations' processes and therefore process modeling is intensively used for process improvement.

Business process modeling is often used as the basis for process improvement, since it enhances understanding of business concerns among various stakeholders [23][24]. The aim of process modeling is to make the processes explicit and provide understanding among the organization even by people with little or no technical knowledge. Studies show that, for this purpose, graph-based visualizations of processes is most useful since they provide a more explicit visualization of the process [25]. On this side of process modeling, the focus is on understandability, therefore the process models are created with high levels of abstractions [26]. When the focus for modeling is improvement, details such as organizational structure and functional operations should also be included in the process definition [27]. These details may not always be related to the process itself but they enhance the understanding among different perspectives and therefore they should be present in the model.

Most often, bottom-up approach to process modeling is used for business process improvement initiations [28]. Bottom-up approach, rather than a top-down one, starts defining the activities at lower levels of operation and continues by combining these activities into higher level of processes. This approach is useful since it gives the modelers detailed insight about the activities performed in the process [29].

2.2 Business Process Management

Before the time of computers, humans were in the center of organizational processes focusing on different roles and responsibilities and the interactions among each other. Although the main considerations are same, emerging technologies have altered the concept of processes enormously. In 1990s, with the emerging of workflow management systems (WfMS) it became possible to link different software running on different platforms. The focus have shifted from people to automation of tasks.

The Workflow Management Coalition (WfMC) defines workflow as automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [30]. Workflow is enacted with workflow management systems, which are systems that define, create and manage the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications [30].

In 2000s, the focus have started to shift again. It became clear that having automation without people was not the best approach since business processes were a combination of people and workflow. This is when the concept of business process management (BPM) has emerged. BPM is using methods, techniques, and tools to support the design, enactment, control, and analysis of operational business processes that involves humans, organizations, applications, documents and other sources of information [30]. As this definition suggests, BPM has roots from both business administration and information systems perspectives. As Nigel [31] summarizes, BPM can be defined as all of the below:

- the definition, recording and use of business rules
- method of gathering management information
- a means of auditing everything a worker does
- a logical progression from automation in the production line to automation in the office
- business process re-engineering
- fine tuning of business processes
- a means of allowing collaboration between workers
- a means of saving money by reducing the load on workers, or even reducing the workers

- an outsourcing method where the work is done by computer rather than offshore worker
- a method for making the processing by one person match the processing by another
- a way of defining and running a business

In its current state, BPM has gained attraction by both business and IT users, since it enhances many aspects for managing business processes. Business users have focused on improving the processes of the organization while reducing the costs of operations, whereas IT users have focused on obtaining scalable and robust software.

BPM has five basic phases in its life-cycle which are listed as follows [32]:

- Design: In this phase, business processes are identified, reviewed and validated. The aim of this phase is obtaining a correct and complete design of organizational processes. As in every task, having a good design in this step ensures a quality process management, reducing the probability of facing problems in the following phases.
- Model: This step defines the processes making them explicit. The processes are documented which is most often in a form of process models.
- Execute: Executing processes is a major contribution of BPM to the organization. Automating the processes increase governance, auditability, accuracy, speed and repeatability of the processes while decreasing learning curve, employee turnover and training time. Automating processes is a very complex task and achieving it involves integrating with many disciplines ranging from decision support systems to neural networks, from business rules engines to bayesian belief networks, etc.
- Monitor: In this step, current processes are measured in order to identify improvement opportunities.
- Optimize: In this phase, what aspects of the processes may be improved is known and it is time to define how these may be accomplished.

These phases clearly indicate that BPM is very closely related with business process improvement. It combines organizational assets such as human resources and legacy systems in order to optimize the value produced by the organizational processes. BPM phases also emphasize the importance of process modeling in BPM life-cycle. If the quality of the process models is poor, the results of following phases will also be poor, leading problems in the later phases [33].

For several years, BPM has been regarded in top ten priorities of CIOs. Gartner and Forrester state that the BPM market is very competitive and no specific product has emerged as a dominant force [34]. The reason for this is there are very different types of challenges and the main players of the sector has no common understanding of what makes a product leader.

There are several industry initiatives, standardization bodies and organizations working on BPM concepts and standardization, including BPMI (Business Process Management Initiative), WfMC (Workflow Management Committee), OASIS (Organization for the Advancement of Structured Information Standards) Committees, Rossetta Net, W3C (World Wide Web Consortium), and OMG (Object Management Group) etc.

With the support from both academic and industrial domains, execution of processes became increasingly popular in organizations. Consequently, many vendors have introduced Business Process Management Systems (BPMS) in order to support BPM life-cycle. BPMS are complex systems with sets of tools to support the various phases of BPM [35]. In 2008, four very large enterprise software suppliers -IBM, Microsoft, Oracle and SAP - made BPM a central element of their product portfolios and marketing efforts [34]. In its current state, The Business Process Management Group lists over 400 vendors fro BPM systems. As Smith [36] states, there are many advantages of using BPMS such as integrating heterogeneous application development environments, combining human tasks with workflow applications, allowing web-service orchestration, enhancing standardized user-interface definitions and monitoring process instances.

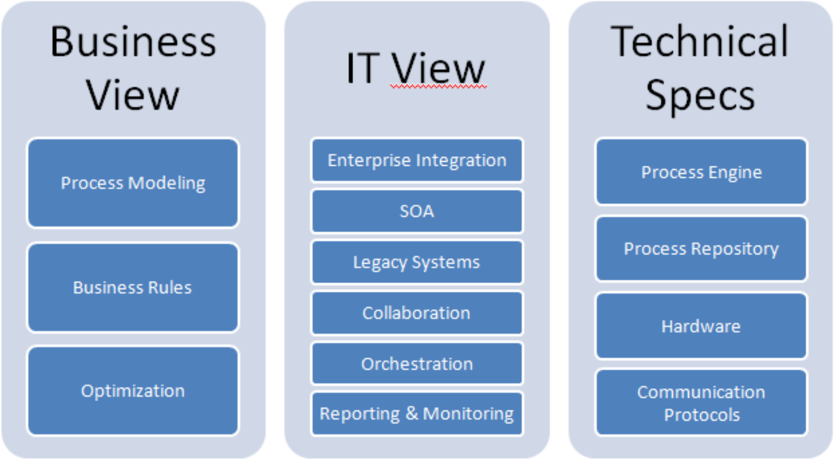


Figure 2.1: BPMS Components

BPMS aim to achieve a very wide spectrum of goals, therefore they have a wide area of interest groups as Figure 2.1 illustrates:

- Process modeling should provide users with a graphical notation to create the process diagrams. These diagrams should include process activities, the sequence, participants, roles, decisions and rules. Users should also be able to define exceptions and error conditions.
- Business rules should be defined using a rules repository.
- Processes should be traceable in terms of performance so that optimization opportunities should be observed.

- Service Oriented Architecture (SOA) provides a modular and reusable approach to software developments and enhances organizational integration.

- Process repository contains deployed processes for instantiation.

- Enterprise systems should be integrated such as databases, messaging systems, organizational legacy software systems.

- Real-time reporting on the current status of process must be available to monitor the business activity.

Due to the complexity of process management systems they are very complex to use, especially for business users. The following aspects of the BPMS make them difficult for organizations to integrate with their systems:

- Modeling Environment: Some BPMS offer either a stand-alone application or an Eclipse plug-in for process modeling. The Eclipse environment does not offer a user-friendly graphics interface and it is not easy to visualize the model, especially for end users who are not familiar with programming interfaces. Although stand-alone applications for business modeling are more user friendly, overall, the modeling environments which BPM suites offer are limited when compared to tools which are specifically designed for business process modeling, such as ARIS [34]. For example, the hierarchical organization of the processes is not permitted in BPM suites, in this case modelers have to deal with the whole process at once, which is not easy to handle.

- Utilization of business process model modeled with a business perspective: Processes modeled in eEPC cannot be reused in any of the BPMS. There are implications that BPMN specifications can be imported into some suites but a one-to-one transformation cannot be achieved. In order to execute processes with BPMS, modelers have to remodel the processes in BPMN notation using the modeling environment that the suites offer.

- Interchange format: XPDL is used in some suites as an interchange standard but problems arise for some elements such as, connectors. Hence, exporting process models from BPMS as well as importing is not free of problems.

- Modeling: BPMS mostly support BPMN notation, however there are some problematic areas. Since the execution perspective is dominant throughout the modeling phase, there are some controls that make it very difficult to express the circular flows in the processes. Another issue is role definitions expressed as lanes in BPMS. Defining role interactions may be quite difficult since each interaction requires the definition of execution parameters. In the business view, the focus is on understanding, therefore elements which are not immediately relevant to processes but enhance understanding are permitted. However, in BPMS models the focus is on automation, therefore, these kinds of elements may be omitted and some knowledge may be lost, not all aspects of the process can be reflected in the BPMS environment.

- Model check: Most suites provide a syntactic check for the notation they support, however, the resulting errors become confusing for business users.

- Executability check: Some BPMS provide executability check during modeling, however, the support for resolving issues is very difficult and modelers need a high level of technical knowledge. For example, Intalio gives error messages for wrongly associated elements but it is necessary to have BPEL knowledge to understand what the error refers to. In other suites, modeling phase is separated from execution phase; therefore execution check is not immediately available.

- Effect of changes in processes: Input/output relations maintain the interactions between processes. Therefore, one change in a process affects other processes only if there are any alterations in input/output definitions.

- Execution Environment: All BPM suites use web technologies to execute the processes via Microsoft, Java or other proprietary solutions. Setting up the environment is time consuming and very difficult, since each suite has its specific requirements, for example, Intalio requires Java, Eclipse and J2EE knowledge to run the server. BizAGI requires familiarity with Microsoft technologies and Web-methods require high RAM and extensive knowledge of the suggested framework.

- Execution parameters: Suites have different approaches for defining execution parameters. Some suites have separate definitions for each phase. Users model the process first, and then, later, add roles, inputs, outputs and relations with other organizational services. This approach is easier since it separates the modeling phase from execution, reducing the complexity of modeling. However, some tools require definition of forms, inputs, outputs while modeling, which requires both programming and BPEL knowledge. For example, there is more work to define conditions for expressing loops and exclusive gateways, and this cannot be done without technical knowledge and experience.

2.3 Business Process Modeling

Business process models help increasing communication among stakeholders and creates a common understanding of the processes [37]. They are crucial inputs for both process improvement and process management, however, most of the current research takes these topics in isolation. As a result, in practice, organizations establish different process models. In process improvement, the focus is on understanding the process and the modeled elements reveal the process together with its organizational concern. They may be subject to interpretation in terms of judgment or expertise of the modelers. However, in process management, process modeling requires to be specific and exact, focusing more on sequences, data definitions, input/output relations and integration with organization's technological infrastructure. Therefore, the produced models are different in semantics and structure.

A large number of graphical process modeling languages has been developed to aid organizations in defining their processes for various purposes. The purpose of modeling has a wide range in a scale from social aspects such as maintaining human understanding and communication to more technical aspects such as executing processes. Similar to the variety in the purpose of modeling, the languages supporting modeling range from descriptions in natural languages to flowcharting techniques and to more advanced languages enabling process simulation and execution. There are significant overlaps between languages, as well as significant differences even among the members within the same family. Understanding the nature of these notations is a difficult task, by itself, however when an organization needs to use process models for process improvement as well as for process execution, the picture becomes particularly confusing. A more complex challenge for organizations regarding process modeling is maintaining consistency and standardization across projects to establish a corporate view. This would be achieved by establishing a methodology by identifying responsibilities and roles, setting metrics and goals enabling comparative evaluation of projects in terms of time, cost or performance and identifying customizations which would enable tailoring the methodology according to project characteristics.

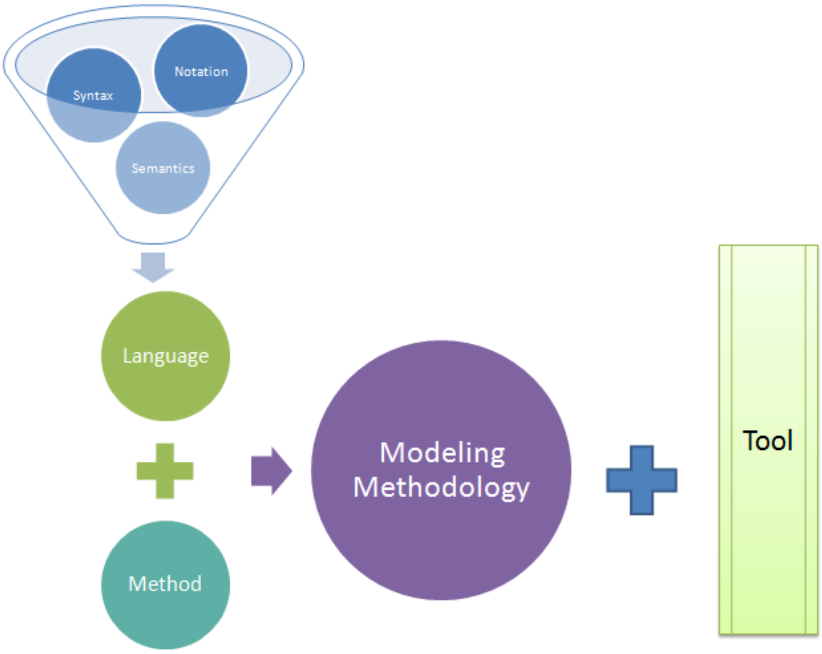


Figure 2.2: Business Process Modeling Components

When modeling is concerned, notation comes forward as the most prominent concept. However, as illustrated in Figure 2.2, there are several other components [38]. Notation describes the graphical representation of symbols which will be used to describe the process. Semantics define the meanings of each symbol in the notation, whereas, syntax describes the rules to combine these. Modeling method is a guideline representing how the notation should be used for maximizing the modeling benefits. Finally, a tool support is required to support and utilize the methodology. Despite the extensive attention on this subject, there are no widely accepted standards, and thus, most of the studies regarding process modeling refer to modeling notations. In this section we give a brief description of mostly referenced notations grouped by their intended use. With the belief that notations are necessary but not sufficient for sharing process models among different scopes, we also give brief descriptions of several process modeling methodologies utilizing these notations for various purposes. In this section, we focus on the separation of notation and methodology. We present several process modeling methodologies and describe underlying process modeling notations with the focus of their use in process modeling and process execution.

2.3.1 Modeling Tools and Notations

The history of process modeling languages shows that existing business process modeling languages come from different traditions and, as such, serve different purposes, represent different things and therefore address different user groups [39]. Process-modeling languages and representations need to present different perspectives, which underlie different aspects of information for analyzing and presenting processes [39]. Accordingly, there are many languages for business process modeling focusing on these different aspects [40]. Since these languages address different process abstraction levels they appeal to different user groups. Lippe et al. analyzed various business modeling approaches and identified that processes of an organization can be grouped in three levels each of which can be modeled with different languages and notations [7]:

- Business processes: EPC, IEM, Business Scenario Maps
- Technical processes: BPDM / BPMN, UML, ebXML, RosettaNet, BPML
- Executable processes: XPDL, WS-BPEL / WS-CDL, UML

BPMN 2.0 specification provides conformance levels enabling process modeling for different perspectives [41][42]. We can use and extend the naming conventions provided by BPMN 2.0 specification and analyze process modeling notations in three groups according to the intended users and specific purposes as Figure 2.3 illustrates [43].

2.3.1.1 Process Definition Notations

The phrase ”to model is to understand” summarizes the essence of this subsection [44]. The main concern of these notations is maintaining understandability by people. They can aid business users, who are actually the process owners, with the level of abstraction with which they can define the real-world processes. These languages are typically not formal, but may be extendable to various informal or heuristic analyses. Commonly used languages in this category include IDEF, Event Process Chains (EPC), Role Activity Diagrams, BPMN and UML.

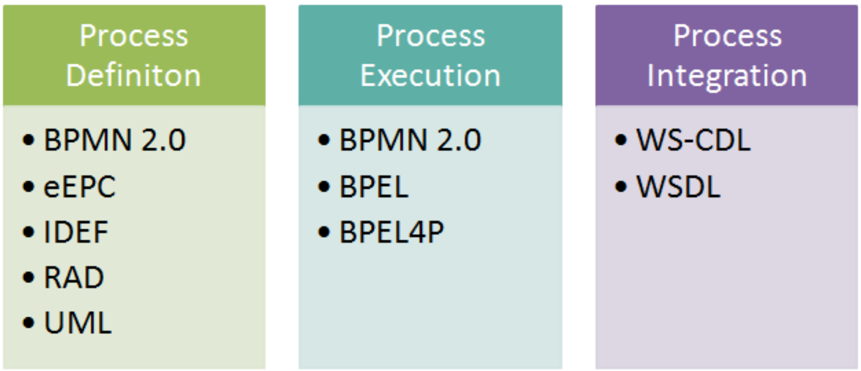


Figure 2.3: Business Process Modeling Notations

1. Business Process Modeling Notation (BPMN)

   The Business Process Modeling Notation is a graphical notation for representing business focusing on the dynamic aspects of business processes; its primary goal is human understandability [41]. It is a standardized graphical notation for expressing business processes. The objective of BPMN is to support both technical and business users. Therefore, it includes simple elements for understanding as well as complex semantics for execution. The BPMN specification also provides a mapping between the graphics of the notation to the underlying constructs of execution languages, particularly BPEL4WS (Business Process Execution Language for Web Services) [45]. BPMN carries out specifications of many other notations and languages such as UML Activity Diagram, IDEF, ebXML, Activity-Decision Flow (ADF) Diagram, RosettaNet, LOVeM, and Event-Process Chains (EPCs). It is maintained by Business Process Management Initiative (BPMI) and Object Management Group (OMG).

   Using BPMN, the activities of the business process and the flow controls are represented on a Business Process Diagram (BPD). The set of elements for business process diagram is explained in detail in Chapter 3.

   BPMN, originating from the process engineering field [46], used for models purposing not only the description of processes but also their execution. Thus it also has a more extensive list of elements that can be utilized for process execution and mapping for execution oriented languages. Figure 2.4 presents an example diagram for the travel booking process [47].

   Support for execution makes BPMN strong in representing behavioral and functional aspects of the processes. However, its support for organization and information perspectives of the processes was weak. The earlier versions of BPMN had problems regarding its meta-model. It was not well-defined therefore it had serialization issues, making it very difficult to transform into other languages. In addition, the role concept, which plays a central part in decentralization of the modeling practice, was not implemented explicitly. The lack of clearly defined semantics and a serialization format led to introduction of BPMN 2.0. In 2010, a beta specification of BPMN 2.0 was released,

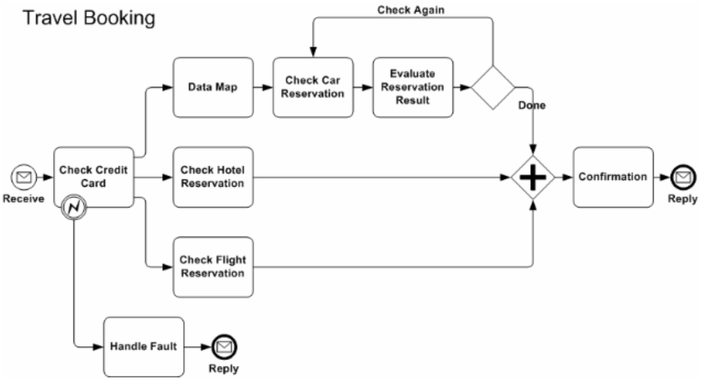


Figure 2.4: Business Process Modeling Notation

formalizing the meta-model. The major objective of this release was a better support for process execution as well as introducing major changes such as:

- New process elements such as non-interrupting events, optional event sub-process, graphical representations of individual task types
- Conversation Diagram to model conversations between participant represented as pools.
- Choreography Diagram to model the sequence of interactions among participants.
- A formal and complete meta-model for definition of model and elements, together with their relationships to enable serialization
- Formalization of execution semantics
- Ability for diagram exchange through XML and XMI.

Another important change introduced with BPMN 2.0 is the definition of conformance levels to support different modeling requirements [48]:

- Process including collaboration
  - Descriptive subclass includes a limited set of elements to enable understanding
  - Analytic subclass adds to descriptive class for more refined requirements
  - Common Executable subclass includes all elements mainly focusing on executional details.
- Process Execution for execution engines
- BPEL Execution for BPEL execution engines
- Choreography for choreography tools

2. Extended Event Driven Process Chains (eEPC)

The Event-Driven Process Chains (EPCs) is used to describe business processes with an informal customer and business perspective. Although it is strong in understandability among business users, it does not support use for formal specifications [49]. extended Event Driven Process Chains (eEPCs) which are EPC's enriched with data, resources, time and probabilities [41] are semi-formal and widely used in the industry. It is a part of the ARIS method and mainly used for business process management, business process reengineering, workflow definition, software development and activity based costing focusing on capturing the organizational view of processes.

The main elements of eEPC are functions and events. Events trigger functions and functions result in events such that a sequence of functions and events represent the process. eEPCs are formed according to the timing of occurrence of each function and event. Control flows are expresses as logical operators. Functions and events can be enriched with the data view of the organizations. Functions can also be assigned detailed sub-models with several levels.

Figure 2.5 presents an example of an eEPC diagram [50].

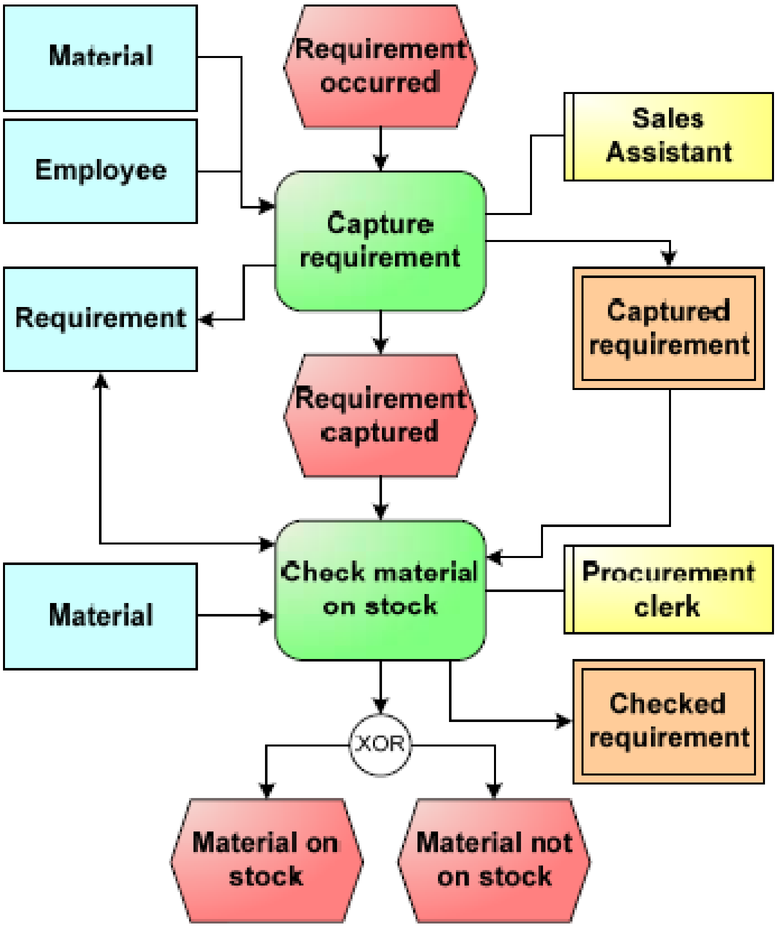


Figure 2.5: EPC

As eEPC is the center of ARIS framework in control view that integrates function, data, organization, and output views, the notation is strong in representing the functional, behavioral, information and organizational aspects of the processes [50].

3. Integrated Definition for Functional Modeling (IDEF)

   The Integrated Definition for Functional Modeling (IDEF) is a family of modeling languages originally evolved from studies in systems and software engineering domains. Although there are more than sixteen versions in the family only the first five of them are widely used [51]. These are:

   (a) IDEF0 - Function Modeling Method

   (b) IDEF1 - Information Modeling Method

   (c) IDEF1X - Data Modeling Method

   (d) IDEF3 - Process Description Capture Method

   (e) IDEF4 - Object-Oriented Design Method

   Of the sixteen modeling languages, IDEF0 and IDEF3 are the most suitable for business process modeling [49]. IDEF0 is a method for modeling business functions and IDEF3 is a method for modeling process descriptions. These two languages are used complimentarily in order to model business processes in an organization.

   IDEF0 modeling language is powerful in functional modeling, which actually is the original objective. A function is an activity, represented by a rectangle, which consumes input to produce some output [52]. As illustrated in Figure 2.6, IDEF0 enables users to represent functions with inputs - resources consumed by the process, outputs - elements produced by the process, controls - objects which guide the process such as policies, standards, etc, and mechanisms - agents which accomplish the tasks [51].

   Unlike many other modeling languages, IDEF0 does not define a sequence of activities. Rather, the relationships between functions define the functional dependencies. This lack of sequencing introduced an advantage in the sense that it allows modelers business to focus on what the process does neglecting the complexity of how it is done. On the other hand, this lack also causes a disadvantage since many people try to interpret IDEF0 as s sequential modeling language such as a flowchart. This lack of sequencing is one of the reasons why IDEF3 is created.

   In contrast to IDEF0, IDEF3 focuses on mechanisms for collecting and documenting processes. Additionally, IDEF3 describes two different types of modeling modes; process flow and object transition network. A process flow description, illustrated in Figure 2.7, captures knowledge of how things work in an organization, e.g., the description of what happens to a part as it flows through a sequence of manufacturing processes. The object state transition, illustrated in Figure 2.8, network description summarizes the allowable transitions an object may undergo throughout a particular process [51].

   Similar to many other process modeling languages, and unlike IDEF0, IDEF3 represents a process as a series of activities. An activity is called a unit of behavior (UOB).

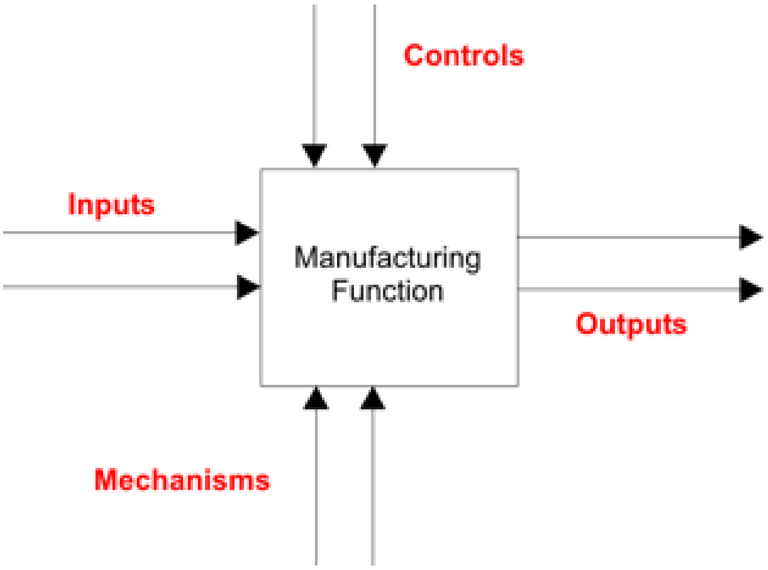


Figure 2.6: IDEF0 Box and Arrow Graphics

The arrows link the UOBs and define the logical flows. Junctions are defined as a mechanism in terms of split, join, branch and merge to define the control flow.

Both IDEF0 and IDEF3 emphasize semantics with an elaborate syntax therefore they are more formal than many other modeling languages. They encourage using functional decomposition.

4. Role Activity Diagram (RAD)

   A Role Activity Diagram (RAD) is an element of the STRIM business process modeling methodology developed by Praxis Plc. for the elicitation, modeling and analysis of business processes [2]. RAD is a role based modeling language which depicts the process as a number of roles interacting with each other.

   A RAD mainly consists of activities, decisions and transactions. The activities are contained in the roles assigned to them. Decisions and transactions are the interactions between the roles. Figure 2.9 shows the basic elements used for RADs [53].

   RADS are very strong in terms of representing role based activities and their interactions since they are aimed to facilitate understanding among participants. However, they are weak in terms of representing informational perspectives of the organization. Moreover, since it has no underlying formal semantics they are not usable for process execution, although there are studies aiming to formalize RADs by translating them to other representations such as Petri Nets [54][55].

5. Unified Modeling Language (UML)

   The Unified Modeling Language (UML) is a general-purpose modeling language, managed by the OMG, originally designed to be used in the field of object-oriented software

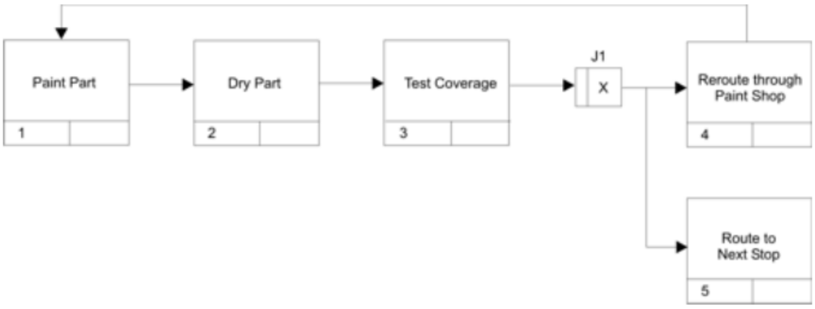


Figure 2.7: IDEF3 Process Description Diagram

engineering. Similar to IDEF, UML also has a large number of diagrams, of which the Activity Diagram is the most used suitable for business process modeling.

UML Activity Diagrams are very similar to RADs, however RAD focuses on roles whereas UML focuses on orchestration of activities.

An UML Activity Diagram consists of action nodes, object nodes and control nodes. An action node is the fundamental unit of behavior specification present in many UML diagrams. In the context of an activity diagram, an action represents some measurable piece of work which should be accomplished by a person or a computer. Object nodes represent the information which is consumed or produced by an action. From a functional perspective, an action is a transformation from a set of input objects to a set of output objects. Control nodes describe some aspect of the flow of control. Important control nodes are initial nodes, final nodes, decisions, forks and joins. The initial and final nodes represent the start and end of an activity diagram. Decisions are used to direct the control flow based on some information. Forks and joins can be used to split and merge the control flow in order to represent parallel processes [56].

Figure 2.10 shows an example of an UML activity diagram [57].

Although the original purpose of UML was to assist the design of object-oriented software systems, more recent versions of UML have broadened its view to systems in general. Particularly, UML Activity Diagrams have been extended with several constructs that are attuned towards the organizational context [56]. While UML was never developed with the goal of business process modeling in mind, it has been used for this purpose extensively. The main motivators for process analysts to use UML are its great popularity, the large breadth of both methodological support and tooling support, the generic applicability of UML for conceptual modeling and UML’s native support for extension with custom constructs. In other words, UML is an accepted and well documented general purpose modeling language.

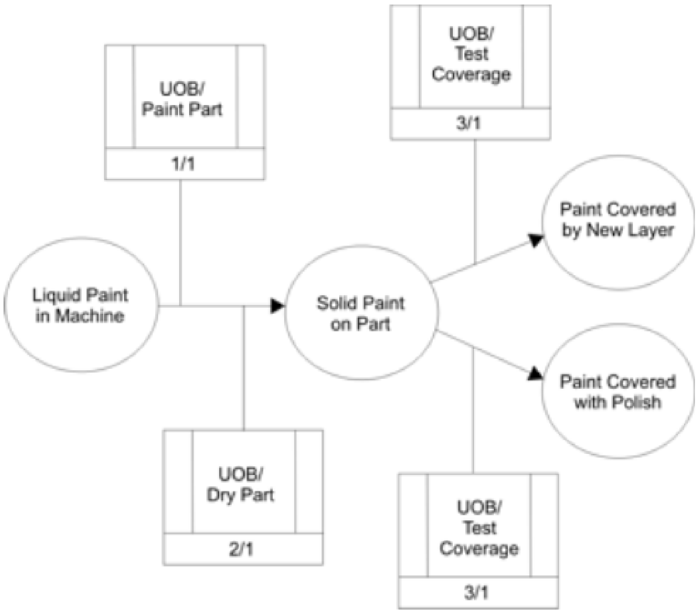


Figure 2.8: IDEF3 Object State Transition Network Diagram

2.3.1.2 Process Execution Notations

Process execution requires engagement of human resources as well as organizations' IT applications and services in simultaneous orchestration. The dominant practice in process execution is using web services. Each activity described in the process model requires software which is generally in the form of web services. These services are deployed in a process engine and are invoked when navigating through the process model during run-time. While the business processes are executed, data and events are generated and they are presented for business process monitoring and further optimization.

1. Business Process Execution Language for Web Services (BPEL)

   Business Process Execution Language for Web Services (BPEL) [45]] is the most popular specification for process execution. BPEL is an XML-based process modeling/orchestration language which is used to define enterprise business processes within web services. BPEL is based on web services in the sense that each business process is assumed to be implemented as a web service. BPEL allows users to define what business process activities interact with web services defined using web service description standards such as WSDL. Since it is focused on execution of models, it is complex and therefore does not appeal to business users.

2. Web-Service Definition Language (WSDL)

   Web Services Description Language (WSDL) [58] is a communication protocol, based on XML, to describe network services as collections of communication points capable

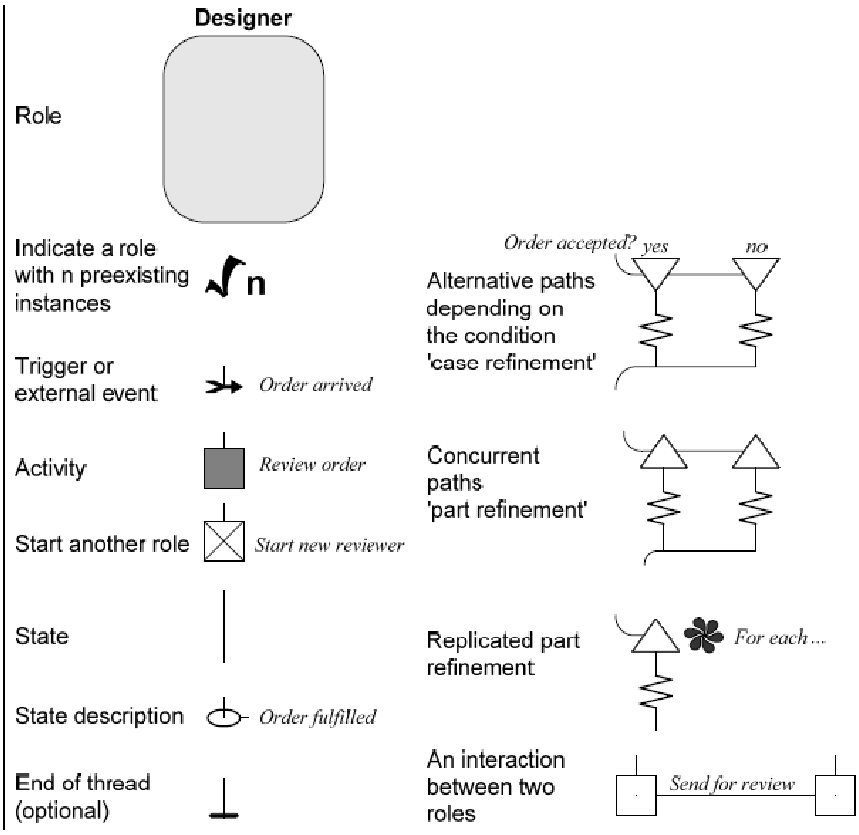

Designer
Role
Indicate a role with n preexisting instances
n
Trigger or external event
Order arrived
Activity
Review order
Start another role
Start new reviewer
State
State description
Order fulfilled
End of thread (optional)
Order accepted?
yes
no
Alternative paths depending on the condition 'case refinement'
Concurrent paths 'part refinement'
Replicated part refinement
For each ...
An interaction between two roles
Send for review

Figure 2.9: Basic Elements of a RAD diagram

of exchanging messages. A WSDL document uses the following elements to define network services [58]:

- Types as a container for data type definitions
- Messages representing the data to be exchanged
- Operation representing descriptions of the actions in a service
- Port Type are abstract collections of operations
- Binding is a protocol and data format specification for a port type
- Port represent the network endpoints
- Service is a collection of ports.

WSDL has become popular mainly due to its interoperability. Many platforms support WSDL in terms interpreting it and making the calls to appropriate web services. However, it is not human readable and very difficult to understand for inexperienced users.

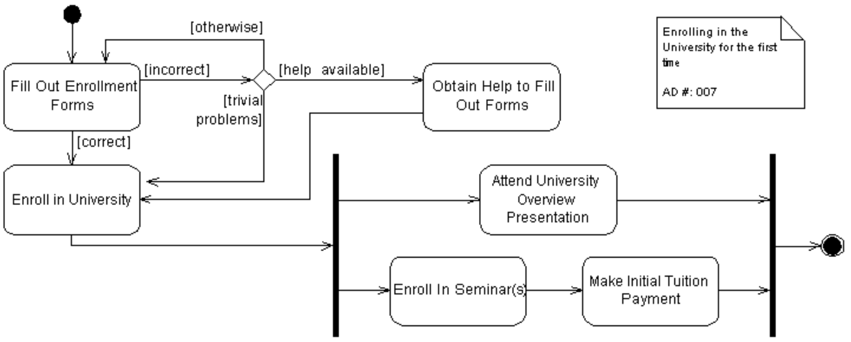

[otherwise]
Enrolling in the University for the first time
AD #: 007
Fill Out Enrollment Forms
[incorrect]
[help available]
Obtain Help to Fill Out Forms
[trivial problems]
[correct]
Enroll in University
Attend University Overview Presentation
Enroll In Seminar(s)
Make Initial Tuition Payment

Figure 2.10: UML Activity diagram

2.3.1.3 Process Integration Notations

These notations aim integrating processes of different business partners. They focus on the mechanics of the integration in terms of abstract, technology independent, programming interfaces and data exchange formats [59]. These languages include RosettaNet, ebXML and WS-CDL.

2.3.2 Modeling Methodologies

Having a clearly defined methodology for accomplishing tasks is important since it outlines directions and provides guidance for procedures to increase consistency. Traditionally, the focus of process modeling have concentrated on notations rather than formally defined rules to guide the construction of a process model. However various studies reveal that modelers get lost in the wide opportunities a notation can offer and require the comfort of formal rules to follow [59][60]. A business process model is the term typically used to describe a particular representation of business process information [61]. The representation is achieved by including the important aspects of the business process with respect to the purposes the model would serve. Thus, a process modeling methodology not only consists of a modeling notation and instructions on how to use the particular modeling notation, but also it includes the concerns for the purpose of process modeling.

Despite the developments in the domain of process modeling notations, the studies focusing on methodological modeling or processes are somehow limited. In this section we present several process modeling methodologies representing different areas of focus.

1. Process Oriented Enterprise Modeling (POEM), formerly known as PRocess EaSY, is a descriptive process-oriented organizational modeling methodology for mainly describing business processes, roles and artifacts [60]. It includes process aspect definitions, which are based on BPMN, and processes, which teach how to conduct process modeling. POEM covers aspects other than process details such as organizational structure,

responsibilities of roles, relations between processes, information exchange between processes, relations between business goals and KPI's and processes, process instances. The potential users of POEM are both process owners and mostly IT staff.

2. In their study, Yamamoto et.al. [62], focus on shortening the time needed to propose solutions to customers regarding business process improvements and system construction or reconstruction. The UML based process descriptions enable process sharing and reuse. The methodology includes preparation and recognition activities to restrict the notation of the modeling result [62].

3. Kim [63] presents EPC modeling which focuses on modeling business processes with EPC notation, including specific cross-functional, customer-oriented business process reengineering characteristics and at the same time attempting to achieve the ideal features of a modeling formalism such as expressiveness, simplicity. The method focuses on process reengineering from a customer specific perspective.

4. Tangible Business Process Modeling (TBPM) aims to get better information and instant feedback from business users and establish a shared understanding of the process by modeling the process together with the domain experts [64]. It includes a tangible toolkit consisting physical BPMN shapes, which increases participation and understanding especially among business users.

5. Amoeba describes a methodology for business processes that is based on business protocols which capture the business meaning of interactions among autonomous parties via commitments [65]. Amoeba includes guidelines for specifying cross-organizational processes using business protocols, which are represented by UML sequence charts, and handling the evolution of requirements via a novel application of protocol composition.

6. PLURAL is a method for organizations to perform business process modeling in a decentralized and concurrent manner [66]. It is based on the idea that organizations' processes can be modeled by individuals actually performing the processes. It offers that individuals are held responsible to model and improve their own processes concurrently, rather than having a central and devoted group of people to understand, analyze, model and improve processes. These individual models are then integrated to form organizationâs process network.

7. Architecture of Integrated Information Systems (ARIS) framework utilizes views in representing the enterprise [67]. The conceptual design of ARIS is a based on an integration concept which is derived from a holistic analysis of business processes perceived with five views; function, organization, data, output and control. ARIS offers various modeling notations such as BPMN, EPC, UML.

8. BIZAGI: In our previous study, we have revealed that various BPM suites propose process modeling methodologies in their proprietary life-cycle [14]. BizAgi business process management suite, for example, offers a BPM solution that enables to model,

automate and execute business processes through a graphic environment and without the need of programming. The process wizard guides the users from modeling to execution with the following steps: modeling the process, editing the data model to identify entities, attributes and relations, defining forms to interact with the users, defining business rules as specific conditions, defining performers in terms of people or resources, integrating with any other system in the organization and executing the process.

9. In his book, Silver proposes a process modeling methodology for BPMN notation focusing on business users and business analysts [42]. He provides guidelines for process modeling, however the focus is on using the BPMN notation correctly and effectively. He gives details on how and when to use BPMN elements to produce correct process models in BPMN notation. He proposes a high-level guideline to simplify and make the process more understandable for business users. Although he states that all relevant stakeholders for the process should be included in the modeling phase, he does not provide any means to check that it is done. His business modeling guidelines should be extended in order to support role-based modeling so that interests of each stakeholder is made certain to be included. Another lacking point in his method is exlusion of execution modeling. Although he provides detailed information on the elements of BPMN mainly used for execution purposes, he does not provide any information on how to include data definitions, user interfaces and organizational integration details.

It is obvious that the approach to process modeling is much more than just adopting a notation; it should include guidance on how, when and who to use the notations specified. As Table 2.1 summarizes, process modeling methodologies naturally evolve in line with the purpose of process modeling. However, there are not many methodological approaches for business process modeling, and there are even less studies focusing on process execution. These studies generally focus on transforming process definition notations to process execution notations [68][69][70]. However, since most process description notations are not formal and do not have meta-definitions, these mappings require preliminary assumptions to overcome the structural differences among notations. A notation may not have corresponding elements in the notation to be transformed therefore their usage may be restricted sacrificing the expressive power.

Not only single elements but also patterns can be problematic. For example business users may model their processes without considering the regard of cyclic patterns which are not easily transformed into execution languages. Enhancing process models with process execution concepts without losing the benefits of graphical modeling notations is a difficult task which cannot be achieved by simply notation transformation. Similar to the ones summarized in Table 2.1, a methodology to produce inputs for process execution using process modeling practices must focus on the attributes of process execution.

In a business model, the elements in the process are described to enhance understanding even though they are not directly relevant, however in modeling for process execution, only related aspects are modeled, those which have no use in execution are omitted. On the other hand, generally, information presented in the business process model is not adequate for a transformation to a complete, executable BPEL process, since they lack the implementation details. The studies on reaching executable solutions from process models have focused on transformation of notations. However, transforming business models into executable languages through notations suffers from some issues. The graphical representations of elements and semantics of process models such as roles, relations, activities, events and constraints, are expressed as variables in transformed execution language. The aspects which are not directly related to execution but more to understanding may be lost during transformation. Due to this loss, data i.e. measurements, organizational structures or rules, needed for process measurement and improvement may be lacking in the executable model. Another issue is since each element may not have a corresponding executable notation, process modeler needs to know the right concepts and elements which will be supported in the executable model. Moreover, the generated executable model may not be usable straightforward after the transformation, since deployment of these definitions on process engine requires engine-specific attributes. When modeling a process, the increase in the abstraction level leads to more complex models. Some processes may be quite large, with many detailed tasks that must be performed by various people. Defining these complex processes for execution is more of a conceptual problem for business people rather than being a technical problem. That is, the complex and large processes must be divided into manageable sub-processes, so that they can be definable by business users. These sub-processes must also be completed with execution, resource, input and output, time and transaction attributes so that they may be understandable not only by humans but also by computers. Moreover, the processes must be monitorable so that the problems such as long completion durations, bottlenecks, resource utilization can be identified and the reasons leading to the problems can be found. Process models for execution must be capable of easily redesigning processes in order to comply with changes, provide documentation for sharing among organization, provide measures for improvement and optimization metrics and easy deployment.

2.4 Re-usability of Process Models Among various BPMS

In order to execute processes a process engine is required. In this section we inspect three BPM suites; BizAGI, Intalio and WebMethods. The focus in this section is not to compare these products, but using these products in order to explore the reusability of process models among different BPMS.

BizAgi offers a BPM solution that enables modeling and executing of business processes through a graphic environment and without the need for programming. The process wizard guides the users from modeling to execution in the following steps: modeling the process using the BizAgi Process Modeler, editing the data model to identify entities, attributes and relations, defining forms to interact with the users, defining business rules as specific conditions, defining performers in terms of people or resources, integrating with any other system in the organization and executing the process.

Intalio—Works is a modular BPM suite that consists of a BPMN modeler, forms editor, data mapper and a deployment manager that is built on an Eclipse plat-form. The suite has a built-in process modeler with BPMN support and can convert the process models modeled using BPMN to BPEL. BPEL is the main means for deploying process models on the Intalio—Server. Intalio has a forum for users to ask questions and get answers, which is very helpful for eliminating modeling problems for execution.

WebMethods provides a web-based platform to design and automate processes, with simulation and monitoring features. The design of the business process management system features an Eclipse based environment in which process models can be developed by multiple researchers in collaboration.

The possibilities of using process models as an input for process execution in BPM suites were identified in accordance with following aspects:

- Modeling Environment: the selected BPM suites offer either a stand-alone application or an Eclipse plug-in for process modeling. The Eclipse environment does not offer a user-friendly graphics interface and it is not easy to visualize the model, especially for end users who are not familiar with programming interfaces. Although stand-alone applications for business modeling are more user friendly, overall, the modeling environments which BPM suites offer are limited when compared to tools which are specifically designed for business process modeling, such as ARIS. For example, the hierarchical organization of the processes is not permitted in BPM suites, in this case modelers have to deal with the whole process at once, which is not easy to handle.

- Usability of model: Each BPM suite requires to remodel the process. There are implications that BPMN specifications can be imported into some suites but a one-to-one transformation cannot be achieved. In order to execute processes with BPM suites, modelers have to remodel the processes in BPMN notation using the modeling environment that the suites offer.

- Interchange format: XPDL is used in some suites as an interchange standard but problems arise for some elements such as, connectors. Hence, exporting process models from BPM suites as well as importing is not free of problems.

- Modeling: BPM suites support BPMN notation, however there are some problematic areas. Since the execution perspective is dominant throughout the modeling phase, there are some controls that make it very difficult to express the circular flows in the processes. Another issue is role definitions expressed as lanes in BPM suites. Defining role interactions may be quite difficult since each interaction requires the definition of execution parameters. In the business model, the focus is on understanding, therefore elements which are not immediately relevant to processes but enhance understanding are permitted. However, in executable models the focus is on automation, therefore, these kinds of elements may be omitted and some knowledge may be lost, not all aspects of the business model can be reflected in the executable model.

- Model check: All suites provide a syntactic check for the notation they support.

- Executability check: Intalio provides executability check during modeling, however, the support for resolving issues is very difficult and modelers need a high level of technical knowledge. For example, Intalio gives error messages for wrongly associated elements but it is necessary to have BPEL knowledge to understand what the error refers to. In other suites, modeling phase is separated from execution phase; therefore execution check is not immediately available.

- Execution Environment: All BPM suites use web technologies to execute the processes via Microsoft, Java or other proprietary solutions. Setting up the environment is time consuming and very difficult, since each suite has its specific requirements, for example, Intalio requires Java, Eclipse and J2EE knowledge to run the server. BizAGI requires familiarity with Microsoft technologies and Web-methods require high RAM and extensive knowledge of the suggested framework.

- Execution parameters: Suites have different approaches for defining execution parameters. Some suites have separate definitions for each phase. Users model the process first, and then, later, add roles, inputs, outputs and relations with other organizational services. This approach is easier since it separates the modeling phase from execution, reducing the complexity of modeling. However, some tools require definition of forms, inputs, outputs while modeling, which requires both programming and BPEL knowledge. For example, there is more work to define conditions for expressing loops and exclusive gateways, and this cannot be done without technical knowledge and experience.

Business process modeling has been a topic of interest for researchers as well as industrial stakeholders since it has immediate commercial benefits in the market. The development of process modeling languages with a wide variety is due, to great extent, to this fact. The notations mentioned in this section are the most commonly used and accepted notations which have evolved from the need of standardization leading to shareable notations among different platforms. Although there are some studies which transform processes in graphical format to process execution notations, the ease of usability of these in practice has been proven false. Process models which hide the increasing complexity of development processes while being precise enough to be executed cannot be simply obtained by notations. Despite the variety of process modeling notations, it is very important to acknowledge that these are only notations; their usage is not defined. Benefiting the reuse of process models through process execution can be accomplished by a complete methodology.

Several gaps occur in business process lifecycle due to lack of communication, especially during the transition from one phase to another phase. Therefore, different models are proposed in literature to fulfill the requirements of stakeholders in business process lifecycle. Therefore, different models and languages are devised to fulfill these gaps and improve the understanding of business processes [65].

Table 2.1: Comparison of Process Modeling Methodologies

Name	Main Focus	Underlying Notation	Intended Users	Modeling Guide	Tailoring Guide	Abstraction Level
POEM	Modeling processes together with organizational concerns	BPMN	Business and IT	Available - as process diagrams	-	High
Yamamoto et al	Business Process Reuse	UML	Business and IT	Available	-	Low
EPC Modeling	Business Process Redesign	EPC	Business and IT	Available - no guide but modeling principles	-	High
TBPM	Business Elicitation	BPMN	Business and IT	-	-	High
Amoeba	Cross-organization Business Process Modeling	UML	IT		-	Low
PLURAL	Process Improvement	EPC	Business	Available	-	High
ARIS	Process Improvement	EPC	Business and IT	Available - generic	-	Low/High
BIZAGI	Business Process Management	BPMN	Business and IT	Available	-	Low
Silver's BPMN	Business Process Modeling	BPMN	Business and IT	Available	-	Low/High

CHAPTER 3

THE PM4E METHOD

Creating business process models that can be used in an organization for different purposes - especially between business and IT - faces many challenges which requires a much more subtle solution than just using modeling shapes and symbols. The main requirement is having a step-by-step guidelines to direct modelers from process descriptions to complete process models. PM4E is a method aiming to accomplish this task. It provides guidelines for organizations to model their processes which they can use both for describing and automating them.

This chapter presents our proposed method, PM4E, in detail. First section discusses the approach in detail with its phases. Second section describes the used notation. In third chapter we discuss on the details of the methodology. Finally, the last section discusses the set of requirements for a tool that can be utilized for the method.

3.1 PM4E Method - Modeling Approach

PM4E is a methodology to fulfill the goal of sharing business models across different audience and platforms with business and IT views. Since there are two main perspectives for this goal, there are two fundamental properties that the produced models should comply.

- In order to maintain understandability and enhance maintainability each process model should have only one interpretation by every user, regardless of the specific characteristics of the users. This property is mostly a concern for the business perspective. They have the domain expertise and the process knowledge and it is mostly their responsibility to define a process model which will be correct, complete and unambiguous.

- Since the process models will be used in execution, they should have only one serialization. The second property mostly applies to IT domain. Once the processes are defined, business modelers do not pay attention to how they will be used by the IT, until a change is required in the process. It is the implementers' expertise to use this models in a way that can be executed directly.

When we look at this separation of concerns, the difference in the purposes of two groups seems so much that, converging their requirements in a single model seems cumbersome [71]. However, when we apply the very basic principle of ”divide and conquer”, we can see that each modeling phase can be continued separately but in harmony utilizing a single model. We divide the modeling process according to business and IT perspectives in the organization and then elaborate on the modeling requirements according to these perspectives. In this way, each perspective focuses on what its core concerns and they are isolated from the parts which may be out of interest.

To accomplish this mission, in PM4E method, we analyze the modelers in the organization in two perspective: business modelers and execution modelers. Figure 3.1 describes the perception of PM4E perspectives in terms of organizational units. First, there are business users who actually carry on the tasks. These people have the process knowledge but, most of the time, they lack modeling capabilities. As a result, most of the time process modeling is done by IT. Second, there are implementers who take in the business models and transform them into software. The software may be created from scratch or it may be implemented as execution of processes in a BPM environment.

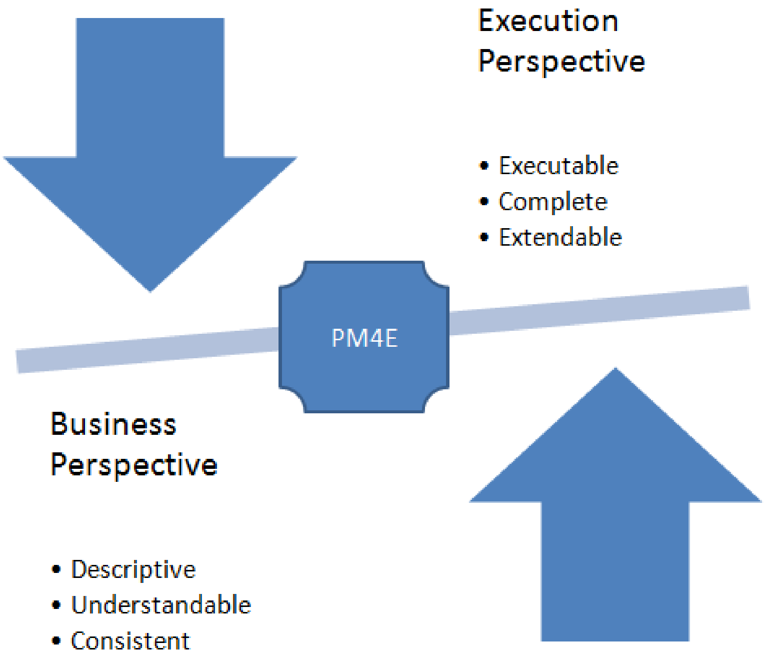


Figure 3.1: User Perspectives in Organizational Structure

PM4E allows each perspective to focus on its own modeling purpose, hiding irrelevant details of the model from each perspective. Since the overall model consists of all the required aspects for both perspectives, there is no need for integration efforts of several separately modeled processes. The resulting process model is complete in the sense that it shelters all the details for two perspectives.

Since we consider the modelers in two perspectives, we apply the same principles to modeling and consider it a two-phase operation.

- First phase is called descriptive modeling which uses a basic set of modeling notation in order to keep the model simple and understandable by any type of users.
- Second phase is execution modeling, which extends the descriptive model by more complex notations provided by the selected modeling notation.

The overall aim is kept intact: having a precise, unambiguous, understandable process model. PM4E is an approach based on encouraging each domain in the organization to model the relevant elements of the process. These elements are kept in a single model in the model repository of the organization, however each domain is presented with a view of the process which is understandable in terms of their experience and knowledge. As depicted in Figure 3.2, the idea is having one single model for each process but presenting it in different ways to different audiences. In this way of process presentation, each participant in the modeling phase deals with what is related to his/her area of expertise and experience level.

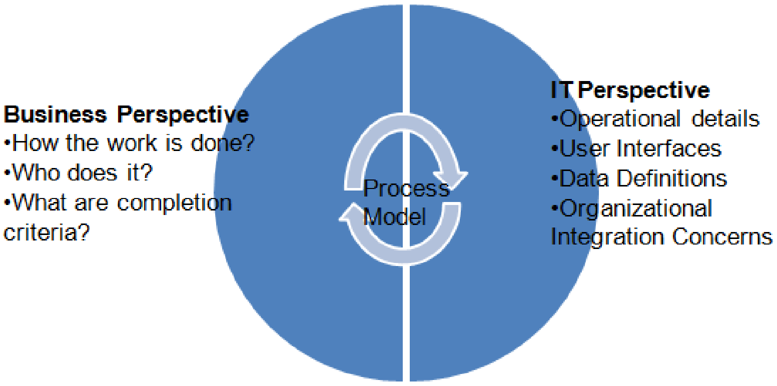


Figure 3.2: One model - Two perspectives

To be able to combine different modeling perspectives in a single model, setting a baseline for the process in question must be the first thing to do. The scope, objectives, roles, main inputs and outputs of what is to be modeled must be decided and agreed on by all stakeholders. Role definitions are crucially important since PM4E is a role-based modeling methodology. After the acceptance of these issues, each modeler is assigned their responsibilities to continue with the modeling process, where the major challenge arises. As discussed in Chapter 2, after inspecting many various BPM suites and methodology, it is seen that current approaches - both academic and commercial - force a single notation for modeling the process throughout the BPM cycle. The notation can be BPMN, SBPM, EPC or some other proprietary notation. However, insisting on a specific notation reduces the efficiency of unaccustomed modelers. Therefore, PM4E gives the users the flexibility of using their preferred modeling notation.

This flexibility, however, issues the question of integration. The concept used in integrating various notations is analogous to adapter design pattern used in software design patterns.

In software development, adapter pattern - or often referred as simply a wrapper - is a design pattern which translates one interface for a class into a compatible interface [72]. An adapter allows classes to work together, which normally could not because of incompatible interfaces, by providing its interface to clients while using the original interface. In the context of PM4E, adapters are used as a method for allowing modelers to switch between different notations.

In current practices, especially when process execution is a concern, the commonly used notation is BPMN2.0. However, eEPC is also very widely used especially by business users since it is very easy to comprehend. In PM4E, the core notation is selected as BPMN2.0 since modeling for execution is an important part of the methodology and BPMN 2.0 has specific execution concerns. We utilize the adapter pattern mentioned above as a means to transfer between eEPC and BPMN notations, as well as between execution modeling languages.

3.2 PM4E Phases

PM4E describes how an organization can use their models both for descriptive and execution analysis. It is a defined methodology, it has its outputs and inputs and it creates value to various stakeholders. Figure 3.3 summarizes the phases in PM4E.

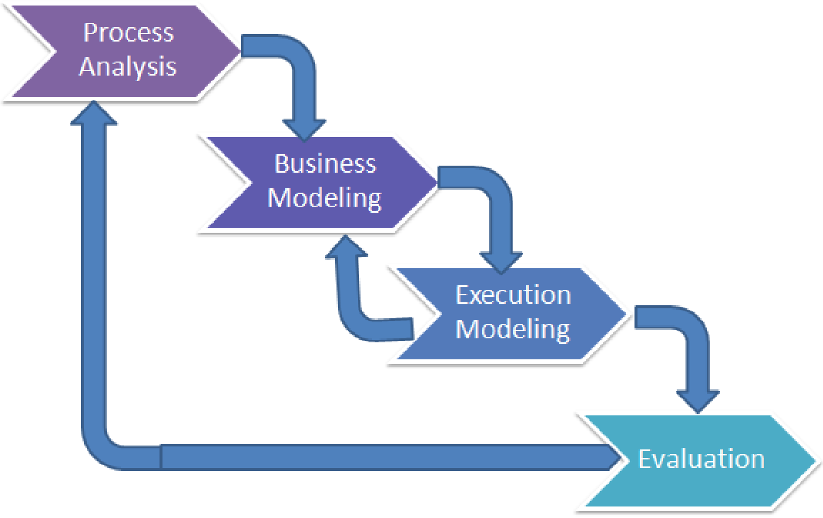


Figure 3.3: PM4E Phases

Process Analysis phase is the initial step where the scope of each process is defined and a basic common understanding among stakeholders is settled. In this phase, the roles are identified and what each role does is defined with required inputs and outputs. After a consensus is settled, business users model the process, with a main objective of describing what is done by whom. The processes are reviewed and once they are approved, IT experts enhance the model with extended elements such that the model could be executed. Then the models are executed and they are evaluated. According to feedback from both perspectives the models may be updated at all levels.

Since PM4E provides executable process models, these models always should be up-to-date reflecting the latest view of the operations in the organizations. Therefore, process modeling in an organization can not be a one-time operation which is completed when the processes are defined, modeled and executed. Rather, it is a cycle of continuously improving operations. After the organizations' processes are modeled and executed, these phases are repeated for every change required for the process.

In the following subsections, the details of each phase will be described in detail together with guidelines.

3.2.1 Process Analysis

As Silver states, process modeling should begin with making all the concepts explicit for each user [42]. In this phase, we follow his guidelines and offer this phase as the initial startup for process modeling. The primary objective is to create a common understanding of what is to be modeled in a high-level view. In Figure 3.4, we can see the main steps in this phase. At first the aim and purpose is defined, then the roles, inputs, outputs and relations with other legacy systems are defined. When this phase is completed, each and every participant should have a clear view of what is to be done and what is expected from them.

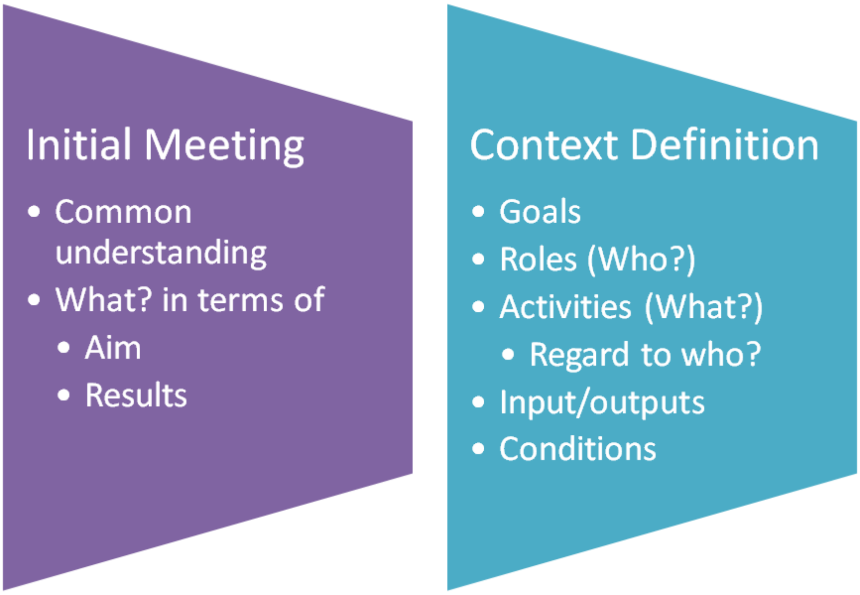


Figure 3.4: Process Analysis Phase

3.2.1.1 Process Analysis Meeting

PM4E process is initiated with an initial meeting. The objective of this meeting is to establish a common understanding of the process among business and IT people. The audience of this meeting consists odf both business and IT groups since they are all owners or the process being implementers and users. Users are usually the business people who actually conduct the process. Implementers are generally people from IT division who will be responsible by executing the processes. They are required in this step because it is assumed that they lack domain knowledge and with this meeting they understand the main concepts and responsibilities fulfilled by the process. In this meeting the following questions are answered:

- What is the scope of the process?
- What are the completion criteria, in terms of success and failure?
- What are the major outputs as well as performance criteria?

Once these questions are answered, the meeting is over and the next step is initiated.

3.2.1.2 Process Definition

In this step the process in question is identified in detail. The audience is formed by process owners and implementers. Using the questions presented in Table 4.4, what? who? when? where? how?, is helpful in the sense of exploring the process. In this step we detail the questions asked in the previous step with relevant users. Process definition step is accomplished by following the subsequent guidelines as depicted in Figure 3.5:

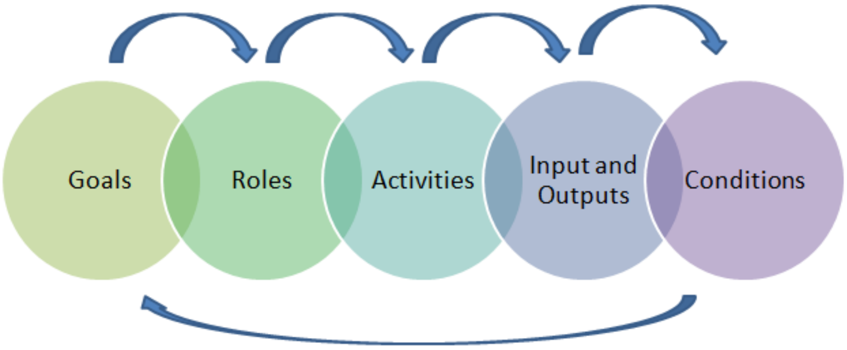


Figure 3.5: Process Definition

1. Identify process goals: Each process must have an objective to complete in order to produce value to its owner. These goals must be explored in this step using the input produced by the previous step. Making the goals explicit helps to define what should

be included in the process and what should not be. They enhance the understanding of the objective by different participants.

2. Identify involved parties (who): Those taking part in activities defined in the processes are expressed with role definitions. Role is defined as a brief description of a person's function in relation to a particular aspect in a process [73]. We extend this description allowing to use non-human aspects of the process as roles. An organizational units, legacy software or hardware are also considered to have roles as well as people. Identifying all the roles in the process in this early phase increases the success of process definition.

3. Define activities (what) in terms of who (roles): Activities may be defined as ordered and structured tasks that are carried out by different roles to achieve a specific goal. Each activity represents a unit of work in the process, identifying the path from a starting condition to an ending condition forming the process sequence. The objective of each task should be identified with the required outputs.

   - Identify the sequence of activities: Once the activities are defined, it is important to elaborate on their sequence.
   - Break down the process to smaller subprocesses: The main objective in this step is to provide understandability, therefore the process should be kept simple. After the activities are defined in the previous step, the ones that are considered as complex should be defined as subprocesses in order to increase readability, understandability and also reusability. Another opportunity for creating subprocesses may arise when similar group of activities are listed as separate tasks. These tasks can be grouped as a subprocess making the process more compact.

4. Identify major input/outputs: Until this step, each activity is described with its inputs and outputs. These may be originated within the process or they may be required from or introduced to other processes, systems or people. Each process must be cross-checked with the previous step and it must be guaranteed that the required outputs are produced.

5. Define conditions. (an email should be sent, you should wait for 5 minutes, etc.): In some cases, there may be specific conditions for an activity to be performed. More often than not, business users forget mentioning these conditions since they are not considered as a part of the process. In this step, each activity is walked through focusing on special cases or requirements.

6. Review process: In this step, the process is inspected in terms of roles, activities, inputs and outputs. The aim is to ensure that:

   - each activity is reachable in the process
   - each activity is assigned to a role
   - each role has at least one activity

- each output defined in initial meeting is produced
- each goal defined by completion criteria in initial meeting is met

The main requirement of this phase is defining easy to understand business processes even for people who do not have experience in business modeling. In this level all details of the process may not be defined. The main objective is having a high-level map of the process definition which is correct but may be semantically lacking. It is also another important aspect that in this phase no actual modeling is performed. Process definition is completed in a hierarchical top-down manner, enabling to add details in child-level subprocesses.

3.2.2 Business Modeling

In this phase of PM4E, the objective is to produce descriptive business process models which are [42]:

- Complete: The major aspects of the process such as how it starts and ends, relations with external entities, the roles and major input and outputs are depicted in the model.
- Consistent: Since PM4E guides through how the process will be identified, all modelers will be able to produce more or less the same output which will result in increased understanding, standardization and consistency among the organization.
- Unambiguous: No ambiguous aspects will be present in the model. The model should be easy to understand even for people who have no modeling experience.
- Shareable between business and IT perspectives: Sharing process models among different stakeholders requires detail and attention. It implies that business users should be more detailed whereas developers understand the process in terms of business functions rather than their specific implementation.

When this phase is completed, the processes should be clear about the focus and objectives. They may be used by process analysts to identify improvement opportunities, therefore they should reveal how the work is done and how can it be done better. They may also serve as the initial inputs for implementation or execution. Although this level still does not provide enough detail for execution, it provides a clarification of some kind of systems requirements. Therefore, process models should be both correct and complete.

It is very obvious that a process model is more than notation. None of the specifications used for modeling provides a methodology for producing accurate process models. PM4E offers a top-down modeling approach which implies the understanding of the overall concepts first and then going deeper in as the analysis of activities require. In this phase, we are trying to avoid bothering business users with unnecessary details, therefore, a top-down modeling approach where details are explored when and if needed is a better way to provide clarity.

PM4E offers the following rules to apply while modeling the process regardless of the notation, in order to achieve non-executable process models which is the objective in this phase. We detail the steps mentioned in the previous section as follows:

- Defining the scope and activities: Once all the stakeholders agree on the purpose of the process, this implies that each participant has the same overall understanding of the process. This is an important aspect in business modeling phase since top-down modeling requires a common consensus on the process to be modeled. The following steps helps ensuring that the scope and basic principles of the processes are understood and made clear by every participating stakeholder.
    - Each process should start with a starting condition. Starting may occur with human interaction or by being triggered with another process.
    - Each process should stop when the completion criteria are met. End events may be more than one, some of them defining success and some defining failure alternatives.
- Modeling: At this point, all the stakeholders agree on what to model. In this step, the process model is defined in a diagramming notation. The following steps, as depicted in Figure 3.6, define how PM4E guides through modeling starting with a high-level description and gradually reaching a complete process model.

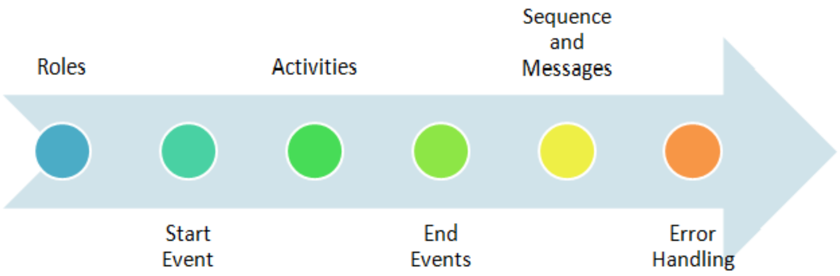


Figure 3.6: Process Modeling

    - Since PM4E tries to align business and IT users on understanding the process models using one single model, it is important to develop main principles for using the modeling environment. Business models produced in this step will be used as inputs for execution step, therefore, it is a good practice to comply with the requirements of the subsequent phases. Process model may be diagrammed either by using a left-to-right or top-to-bottom orientation. Organizations should decide on the orientation and use a standard view in order to increase understandability. We use a left-to-right representation since they are believed to be more readable by humans.

- The first step in PM4E business modeling is to represent the roles. By addressing the roles, we ensure that each ”who” in the process is placed in the process model.
- Starting event should be the first element to be placed in the diagram together with its role, i.e. who/what starts the process.
- From the previous phase we have a list of major activities carried on in the process, with the knowledge of who performs it, what are the inputs and what are the produced results. We place the activities to relevant lanes with their inputs and outputs. When there are decisions reflecting alternating paths, relevant type of gateways must be placed between the activities. An important consideration is the number of activities in each process diagram.When there are a large number of activities the process model is perceived as cumbersome and therefore it is less maintainable. Keeping the number of elements minimum for each process is also useful in the execution modeling phase since each element will be overviewed. When there are a large number of activities, they should be grouped in smaller sub-processes and diagrammed separately.
- A process is a sequence of activities. The start of an activity is triggered when a previous activity in the process is completed. Therefore, the activities should be placed in order of execution. The order of the activities are made explicit by sequence and/or message flows. The decision points, which are expresses as gateways, should also be explicitly represented while placing the activities.
- Many processes have more than one ending conditions. After the activities are placed in order, we should continue with the end events.
- At this point, the process model is constructed. As the last improvement, exceptions must be identified and the model should be extended with additional activities or end events.

- Review the model. Modelers should check that:
  - all the roles identified in the process description phase are represented in the diagram.
  - all activities are connected with roles
  - all input/outputs defined in the previous phase are connected with activities.

When the organization has no specific business process modeling notation or it already uses BPMN, PM4E offers to use BPMN2.0. However, when the organization uses other notations, we encourage to use their preferred notation in order to reduce the resistance to participate in modeling efforts. In the next sections, we describe in detail the usage for BPMN2.0 and adapter mechanisms for other modeling notations.

3.2.2.1 Modeling with BPMN2.0

BPMN, which used to stand for Business Process Modeling Notation until 2011 when BPMN2.0 is introduced, is a diagramming language for depicting business process models. The fact that it has become a major standard utilized by many organizations is mainly due to the reason that many vendors support this standard. Especially OMG's support for BPMN accelerated the acceptance of this standard among organizations as well as tool vendors. Before version 2.0, processes modeled with BPMN could not be executed directly in process engines, since it did not cover all the technical aspects required for execution. However, with the introduction of BPMN2.0, direct execution of BPMN models on process engines became possible. This is made possible by both introducing a set of symbols and providing a methodology including the usage of syntax and semantics. Therefore, BPMN now stands for Business Process Model and Notation, since it provides both the meta-model for semantics and serializability.

In this phase, we recommend the use of elements presented in 'Descriptive Process Modeling Conformance Subclass' by BPMN2.0, which are listed in Table 3.1 [41]. These elements are more than enough for business users to represent the logic of the process without dealing with the complexity of many other elements introduced with BPMN2.0.

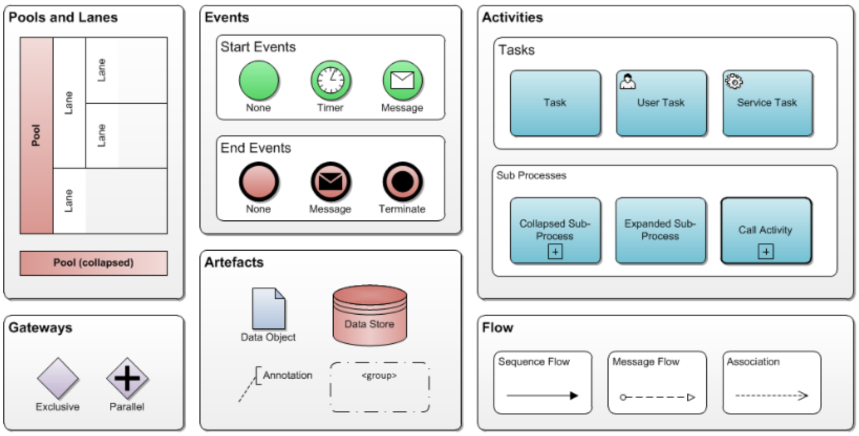


Figure 3.7: Representation of Elements

The above Figure 3.7 depicts the representation of the above elements in BPMN2.0 notation. In this section we will give detailed information about the usage of these elements.

Activities are major components in a BPMN model. They define the work performed by a specific role. Activities are either tasks or subprocesses. A task represents an action with the following extensions:

- User Task: a task performed by a person
- Service Task: an automated activity initialized without human intervention
- None: an undefined type of task

Table 3.1: Descriptive Process Modeling Conformance Subclass

Element Group	Name	Attributes
Activity	UserTask ServiceTask Task(None) Subprocess CallActivity	id, name id, name id, name id, name, flowElement id, name, calledElement
Gateway	exclusiveGateway parallelGateway	id, name id, name
Start Event	startEvent(None) messageStartEvent timerStartEvent	id, name id, name,messageEventDefinition id, name,timerEventDefinition
End Event	endEvent(None) messageEndEvent terminateEndEvent	id, name id, name,messageEventDefinition id, name,terminateEventDefinition
Sequence Flow	sequenceFlow	id, name, sourceRef, targetRef
Message Flow	messageFlow	id, name, sourceRef, targetRef
Pool and Lane	participant laneSet	id, name, processRef id, lane with name, childLaneSet, flowElementRef
Data	DataObject dataStoreReference association	id, name id, name,dataStoreRef id, name, sourceRef, targetRef, associationDirection
Documentation	documentation	text
Artifact	TextAnnotation Group	id, text id, categoryRef

A subprocess, which is another type of activity, represent compound activities. There are also start and end events in a subprocess and when a subprocess is reached in a process, the flow starts from the start event and escalates to the end event and continues with the subsequent task in the main process. Subprocesses are helpful in the sense that they make the complex processes more readable. BPMN2.0 distinguishes a subprocess from a Call Activity in terms of their reusability. If the subprocess is to be used in another model, then it is better to define it as a Call Activity rather than a subprocess. In terms of business process modeling from business users' perspective, however, both subprocesses and call activities are complex structures reducing the readability for business users. Therefore we propose to separate the processes rather than using these elements.

Gateways are used to define the control flow in the process among different alternative paths. There are two types of gateways used in this phase:

- Exlusive Gateway: It is most commonly known as XOR gateway meaning that only one of its outgoing sequence flows will be enabled in any instance.
- Parallel Gateway: It is also known as AND gateway meaning that all of its outgoing sequence flows will be followed in parallel.

Start events indicate where and how a process starts. Normally, each process has one start event which may be one of the following types:

- None: No trigger is defined for the start event meaning either the trigger is unspecified or a task performer starts the process
- Message Start Event: The process is triggered after receiving a message from outside the processes.
- Timer Start Event: The process is usually triggered with a defined schedule.

End events define the completion of the process. There may be more than one ending conditions in a process and each one of them must be defined explicitly using following notations:

- None: No task is required when the process is completed
- Message End Event: The process is sends a message upon reaching the end state.
- Terminate End Event: The process is terminated even if there are other instances running.

Sequence flows displays the sequential order of execution of the tasks in a process, whereas message flows represents the communication between process and external entities.

Pools represent different participant within a process and they may contain several lanes which are the roles or functions that group a set of activities within themselves.

Data objects represent which data is consumed in an activity or which data is produced as a result of an activity.

The tool support for PM4E business modeling method is developed using BPMN 2.0, therefore, our method is clearly applicable for using BPMN2.0 in business modeling phase.

3.2.2.2 Modeling with eEPC: Using Adapter

As mentioned in previous chapter, Event-driven Process Chain is a component of ARIS methodology developed at Saarland University in cooperation with SAP AG. ARIS Toolset is utilized by many ERP solutions provided by SAP, which is one of the major reasons why EPCs are used very intensively. Another reason why EPCS are widely used is the fact that they represent the relations with other elements of the enterprise architecture organizations very powerfully. Since they are very powerful for expressing the needs of business users, how they may be used in this phase is explained in this subsection.

The eEPC mainly consists of three basic symbols which are functions, events and connectors as well as symbols to define organizational units, data and application systems. Considering the organization would prefer to process their models using eEPC notation, we should define an adapter to convert eEPC models into BPMN notation. The conversion mechanism is defined as follows:

- In EPC a function defines a basic unit of objective in terms of the business process. It has inputs which may or may not be transformed into outputs. In BPMN, tasks are the elements that corresponds with this element of EPC. Therefore, functions are transformed to tasks.
- Connectors in EPC are connection points in the process for events and functions. Corresponding elements in the BPMN notation are gateways. The transformation should be done mapping the connectors to gateways basing on their split or join behavior. In Figure 3.8, notation transformations are displayed [74].

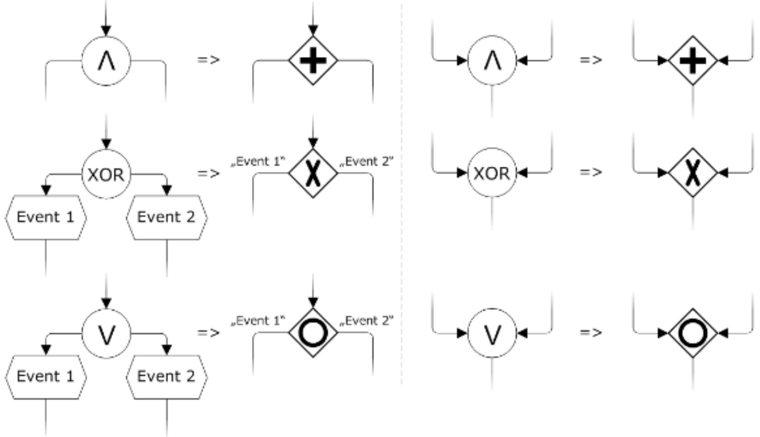


Figure 3.8: eEPC - BPMN Conversion

- Events in EPC are not easy to map to BPMN events, because although the names are same, the specifications are quite different.The difference is, an event in BPMN is defined more as a trigger or a consumption of something which causes or has an impact on the following process flow. Hence, it is not possible to map an event in EPC directly to an event in BPMN. Although there are difficulties, the following rules make it possible to convert events.
  - An event with no incoming message is a start event
  - Intermediate events in the eEPC diagram are unnecessary in BPMN therefore, they are neglected.

  - An event with no outgoing message is an end event

- An organization and a position in EPC are mapped to a lane in BPMN which is a child of a pool.

- Data objects can be mapped directly to each other.

eEPC has lots of users and there are many tools and studies for transferring eEPC notations to BPMN notations. One of these tools can be used as an adapter for input to be used in the following steps of PM4E method. However, it must be made sure that the models are consistent with each other.

At this point in the PM4E process, business modeling phase is completed. The activities and major objectives in this phase can be summarized as follows:

- In this first level of modeling, the objective is to make the process in question explicit by defining it in a complete, consistent and an unambiguous manner.

- The participants in this level have domain knowledge and expertise, but they are not required to have skills in process modeling. Therefore, the modeling process is kept as simple as possible with the following steps:

  - Define what is included in the process and what is not
  - Define roles
  - Model starting and ending conditions
  - Model activities, their order and decision logic together with inputs and outputs
  - Extend the model with exception conditions
  - Review the model

- After the model is completed, we have the process definition in a model that can be used for input for execution, or can be used for descriptive purposes such as understanding the process, teaching the process or improving it.

3.2.3 Execution modeling

In this phase of PM4E, the objective is to produce executable process models using the previously designed high-level process definitions. The recipient of the output of this phase is a process engine, therefore the process model should be correct and complete both syntactically and semantically. This model will describe how the process is actually conducted. The modelers in this phase will most likely be the IT experts because high level BPMN and system design knowledge is required to be able to produced complex business models which would be interpreted by process engines. Going from a high-level process description to a

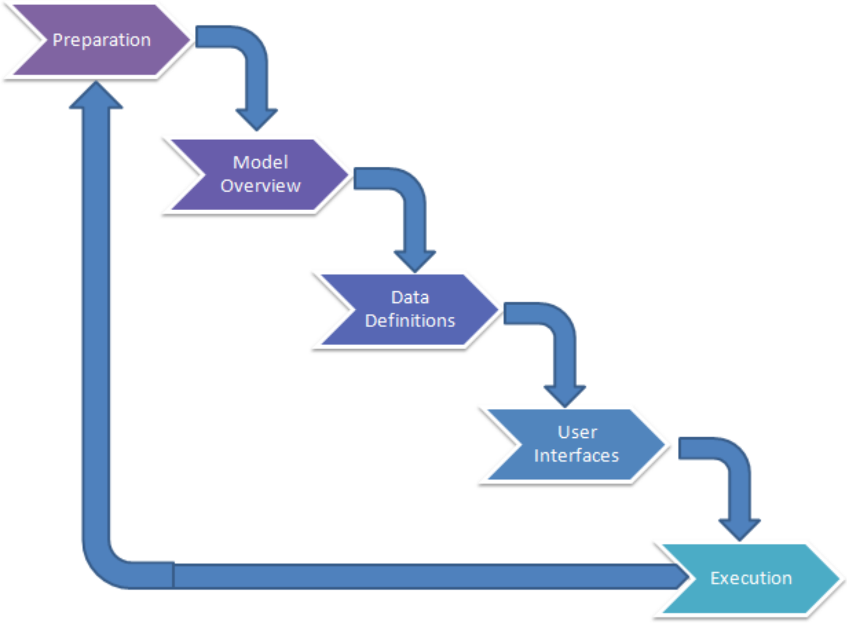


Figure 3.9: Execution Modeling Steps

very detailed process model requires several steps as presented in Figure 3.9. PM4E execution modeling offers iterations on the previously built business model in order to achieve an executable process.

In this section, the details for each step is described in detail.

3.2.3.1 Preparation

Until this point, how the process will be actually executed in organization structure was not a consideration. At this phase of PM4E, we will focus on how these models can be executed in process engines and the requirements to accomplish this task. Although how different process engines operate are similar as depicted in Figure 3.10, each process engine has its own implementation therefore its own specifications. The choice of the engine has impacts on te execution modeling steps. However, BPMN2.0 is designed to be an interchange format to produce directly executable models according to various specifications.

Before starting modeling for execution, the very first thing that should be decided is the technology and language to use. If the organization does not have an active process engine or a BPM product then the technology and the process engine to continue with should be decided. Afterwards, the model should be detailed according to this engine. PM4E guides users to focus on requirements needed to extend the model for execution as identified and described in the following sections.

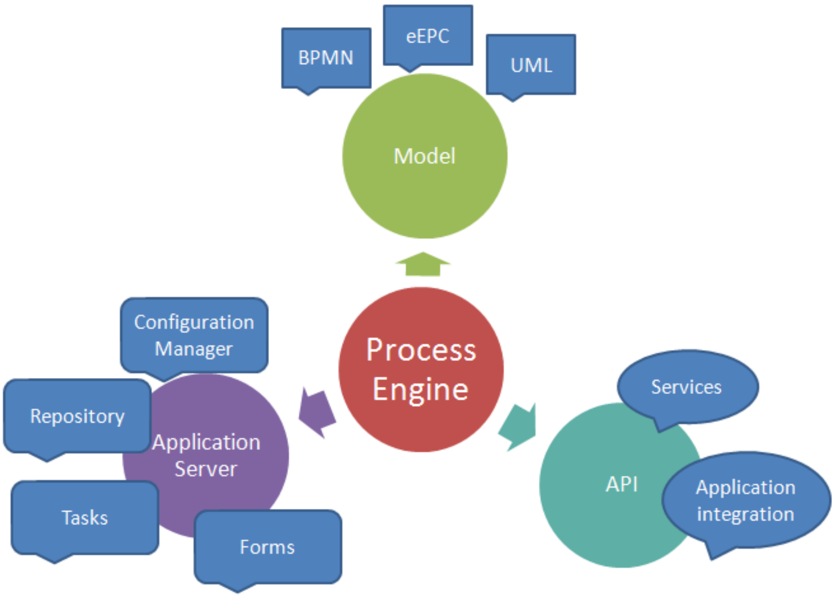


Figure 3.10: How Do Process Engines Operate?

3.2.3.2 Model Overview

Business modeling is performed with a limited set of BPMN2.0 elements. These limited elements are sufficient for business users to express their intentions and define the process, however, for execution modeling no restrictions should be defined on the element list. Each activity, event and gateway should be inspected in order to find their most appropriate element type, as summarized in Table 3.2.

The executable model starts to form with the beginning of this iteration. Extending the business model's elements to include more complex BPMN2.0 elements requires extensive knowledge of task and especially event types. Therefore, this step is carried on by IT experts. However, business users should be consulted for each element overview, since, to be able to decide what element type to use, IT experts should know about the details of each task, event or decision points and this information is held by the business users.

PM4E offers an initial meeting for business users and IT experts. In this meeting, each element of the process in question is overviewed and the details are discussed. For each elements our questions from Chapter 2 will be helpful to dig in the details required by the element:

- What will be produced after the responsibility of this element is fulfilled?
- Who will be informed when this task is completed?
- When will this task be completed? Are there any other organizational restrictions or criteria to be met?

Table 3.2: Analytical Process Modeling Conformance Subclass

Element Group	Name	Attributes
Activity	UserTask	id, name
	ServiceTask	id, name
	Task(None)	id, name
	sendTask	id, name
	receiveTask	id, name
	Subprocess	id, name, flowElement
	CallActivity	id, name, calledElement
	LoopingActivity	standardLoopCharacteristics
Gateway	exclusiveGateway	id, name
	parallelGateway	id, name
	eventBasedGateway	id, name, eventGatewayType
Start Event	startEvent(None)	id, name
	messageStartEvent	id, name,messageEventDefinition
	timerStartEvent	id, name,timerEventDefinition
End Event	endEvent(None)	id, name
	messageEndEvent	id, name,messageEventDefinition
	terminateEndEvent	id, name,terminateEventDefinition
Sequence Flow	sequenceFlow	id, name, sourceRef, targetRef
Conditional Sequence Flow	sequenceFlow	id, name, sourceRef, targetRef, conditionExpression
Message Flow	messageFlow	id, name, sourceRef, targetRef
Pool and Lane	participant	id, name, processRef
	laneSet	id, lane with name, childLaneSet, flowElementRef
Data	DataObject	id, name
	dataStoreReference	id, name,dataStoreRef
	association	id, name, sourceRef, targetRef, associationDirection
Documentation	documentation	text
Artifact	TextAnnotation	id, text
	Group	id, categoryRef

- Where is the actual task performed? Are there relations with other systems?
- How will the process engine decide on if the task is accomplished or not?

When these questions, and new questions raised by the answers are answered and a full understanding of each element is reached, IT experts can decide on which element type is more suitable for modeling for execution. An important point is the fact that the more the execution model becomes complex, the more it scares business users. It becomes less understandable especially because of more types of events. Therefore, in these meetings users must be communicated via the business model. It is a good practice to hide the executional details from business users.

3.2.3.3 Data Definitions

When we recall the business model, there is no mention of data specific attributes such as data types, data structures, the flow of input and output data or the timing of this data flow. However, in an executable model process data is the most crucial variable to address. Although BPMN does not provide data definition standards, it offers extension points by which data can be defined using XML schemas, Java or .NET. PM4E offers to utilize XML Schema definition.

In this iteration, we define the details of input and output data and extend the process elements with these definitions. While elaborating on data definitions, we propose following the process flow in order. This is an effective approach because most of the time the output of an element is an input for the subsequent elements. For every input/output in an element we define the following attributes:

- Name: This is a unique name to identify throughout the process. Even a seemingly very simple process may have a large number of input and output parameters. Managing these parameters become troublesome unless a standard naming convention is used. PM4E offers to use a naming convention such as ElementName:Input_InputName.
- Data Type: BPMN2.0 supports data types used in Java or .NET. However, restricting the modeler to use this style may cause problems with the process engine. PM4E offers to define an XML schema type definitions and using this type definitions for the entire organizations. This will ensure that organization's data definitions will be standardized, since each process will use the same data type definitions.
- Data Mapping: Producing executable process models requires a detailed analysis of data flow as in developing a software. Although the objective for PM4E is not producing zero software, the facilities BPMN2.0 offer enables reducing the amount of code produced. Data mapping enables to identify how the input/output will be operated.
  - The data in question may be mapped to another data in the process, i.e. one element's output is another element's output.

  - Data may be populated by evaluating a transformation or an expression. This actually means that we can produce the data using other data defined in the process.
  - Identifying default values for the data, if there is any, is a useful practice in data definitions, since, they have a positive effect on reducing exceptions.

Once all the input and output data is defined, the process has a complete data model. Data model is kept within the BPMN structure in XML format, however they are not visible to the modelers.

3.2.3.4 User Interfaces

User interface definition is one of the integral and important parts of process execution modeling. It is considered as a part of implementation. The process flow is maintained through the user interface descriptions while executing, therefore it is not easy to separate user interface modeling from the requirements and specifications of the process engine on which the processes will be executed. Each process engine requires deployment of user interfaces in its specific format, which is again an area to use adapter mechanism to export a compatible format with the selected process engine. PM4E does not provide a specific format or notation to use for user interface design, however it provides guidelines to direct the modelers on how to extend the business model to produce the necessary input for user interface modeling.

User interfaces are designed on user tasks since interaction from users are required only for these tasks. The data, inputs and outputs defined in previous phases of modeling are the input for user interface design, which will be transferred into execution variables. PM4E offers the following steps in order to relate tasks, inputs, outputs and the data used within the process as follows:

1. Select the tasks which require user interaction and list them in the provided template, depicted in Figure 3.11.

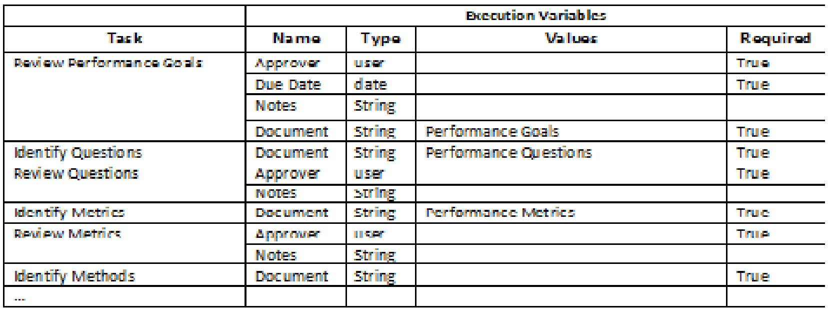
2. For each task identify execution variables as follows:

   - Identify the name of the information which will be inputted from the user. Input and data descriptions may be used in order to embrace all the previously defined variables. These names will actually be seen on the user interfaces in the actual execution of the process, therefore they should be given logical and understandable names.

   - Provide the data types for these inputs. Data types may be generic Java types such as ”String”, or types specific to the selected process engine, such as ”user” in Activiti. PM4E recommends to construct a dictionary in the organization to be able to use these types uniformly throughout the organization.

- Provide default values for the variables. This practice becomes helpful especially for tasks which are not completed in the BPMS. For example, a task produces a document and then it is used as an input for another task. The general practice for handling this is to enable the users to prepare the documents in their word processor, or document management system or any other organizational system and then let them approve that the document is ready in the BPMS so that the process engine may escalate to the next task. For such practices, giving the default values for the document names, i.e. Performance Metrics Report, enhances the standardization among the process models and the execution variables.
- Identify whether the variables is a must field or not. As in every software development system, some variables are obligatory for the operation to continue whereas some others may not be, however they are inputted for other purposes such as performance measurements.

We provide the following template, depicted in Figure 3.11, to use for defining execution variables.



	Execution Variables			
Task	Name	Type	Values	Required
Review Performance Goals	Approver	user		True
	Due Date	date		True
	Notes	String		
	Document	String	Performance Goals	True
Identify Questions	Document	String	Performance Questions	True
Review Questions	Approver	user		True
	Notes	String		
Identify Metrics	Document	String	Performance Metrics	True
Review Metrics	Approver	user		True
	Notes	String		
Identify Methods	Document	String		True
...				

Figure 3.11: Execution Variables for User Interfaces

Once the template is filled for each task, execution modelers are ready to integrate this information into the model itself. PM4E uses Eclipse BPMN2.0 modeler and extends it to include these information. The details are described in the next sections.

3.2.3.5 Business Rules

Some organizations, involving in business process management systems, are also engaging in software systems to define and deploy their business logic, especially when they are complex, named as Business Rule Management System (BRMS). BRMS provides increased control over the decision logic of the organization separating the changes from the IT implementation, however, it requires extensive knowledge on Object Oriented Analysis and Design principles as well as rule definition and integration with organizational systems. Since this subject is yet to be developed, there are no standard implementation rules and meta-models for defining rules within a BRMS, therefore serialization of these systems is not consistent, making it not possible to use adapter mechanism for business rules since each model may have more than one serialization.

Since BRMS are not mature enough to integrate with process engines, PM4E leaves it out of scope and offers to use business rule task provided by BPMN 2.0 notation, if rules are required to be employed in the system.

3.2.3.6 Using Organizational Systems

Executing business process requires two main points of integration with organizational systems.

1. Role definitions: Many organizations have their authorization systems in order to determine whether a person is allowed to perform a requested action. They are closely related to authentication systems which determine if a person is actually the one he/she claims to be. These security aspects are very important for organizations and the processes should be integrated with these systems, otherwise separate authorization mechanisms should be employed in BPMS. The main entity to integrate process models with authorization systems is role definitions. However, current practice for BPM systems is employing their own user management and security modules. It is an active topic for research, therefore, PM4E leaves role integration out of scope. However, it recommends to construct a policy/role definition dictionary in the organization and use consistent naming conventions with the process models.

2. Integration with organization’s other systems: Each organization has its own systems and BPMS should be able to integrate with them . This integration may be at several different layers, such as sharing data from a database, utilizing previously built software using SOA, using organizational e-mail or communication systems, using LDAP systems to update user information or using content management systems to integrate with uploaded documents of the organization. Each of these subjects is a field for special consideration, and similar to role definitions they are not mature enough. The only exception may be using SOA principles to invoke web-based services (SOAP or RESTful services) to communicate between BPMS and other systems in terms of simple data types (i.e. Integer, String) or more complex structures such as XML documents. Since most BPMS rely on SOA principles, they provide integration mechanisms for web-based services, however most BPMS take this integration part optional. Therefore, PM4E leaves this section out of scope.

3.2.3.7 Execution

Most BPM vendors support BPMN however the extend of this support is mainly limited to its notation. Since process execution requires many disciplines to work in harmony, most of the large vendors in this market ignore the executability of this standard and provide their own proprietary solutions. As described in Chapter 2, several transformations may be required for process models to be executed in process engines. For example, Intalio requires process models in BPEL format and web-services and user interfaces in WSDL formats. Activiti, which is the selected execution engine in this study, requires BPMN20 format. In this step, PM4E tool provides a transformation from the BPMN2.0 diagram to BPMN20 format. These process descriptions are then deployed in the Activiti server and the processes are instantiated.

3.2.4 Evaluation

Process modeling does not stop once the processes are executable in the process engine. Since PM4E offers process models which reflect the most current status of the processes, it is a never-ending cycle enabling continuous improvement of the processes. The initial step in evaluation is validate if the executed model works as intended. When there are change requests upon the process either from IT or business departments, the effect of these changes are depicted in Figure 3.12.

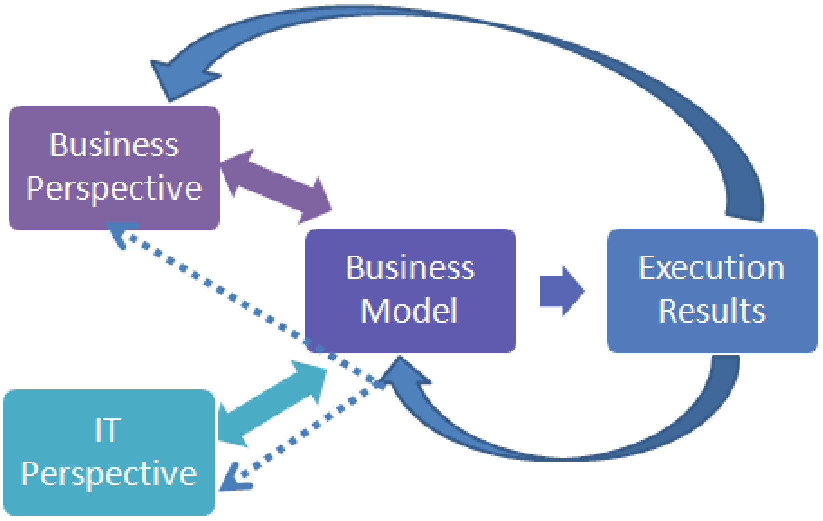


Figure 3.12: Keeping the model up-to-date

Due to results obtained from the execution, the process model may be updated by IT. In this case since there is one single model for all perspectives, the business people will be able to see the updated model. Execution results may be analyzed by business users and change requests may be performed on the model by the business users. In this case, PM4E phases will be carried on to reflect the changes to the executable model and then to process engine deployment. In either way, the process model will maintain its up-to-date status.

3.3 Modeling Notation

PM4E is a guideline to produce executable models from descriptive process models. It requires a notation to describe the processes as well as related execution variables and transformations. This section describes the notation utilized in PM4E method.

3.3.1 Criteria for Notation

In Chapter 2, several process modeling notations and languages and their strengths and weaknesses are listed. PM4E is a process modeling methodology which joins business and IT perspectives in a single model. In the simplest sense, process models try to answer the following questions [39] [14] [75]:

- what will be achieved?
- who will perform it?
- how will it be achieved?
- when will it be done?
- where will it take place?
- why will it be done?

The purpose of modeling requires different levels of answers for these questions and if the modeling notation can cover the requirements for different levels and purposes, then the produced models would be complete and consistent. PM4E integrates two different approaches to process modeling. One is the descriptive approach which focuses on understanding how the work is actually conducted [75]. The other approach is for process execution and it requires to extend the model with specific process execution variables. Process modeling notation for PM4E should be suitable both for descriptive process modeling, which would be used by business users, as well as execution modeling, which would be used for IT experts in order to execute the processes.

PM4E enhances the use of process models especially with regard to following aspects:

- Providing a common understanding of the process among various stakeholders
- Enhancing process improvement

- Executing the process
- Monitoring and evaluating the execution results

The first two aspects are generally the concern for business users, the latter ones are mostly related to implementation which is a concern for IT. A notation which will be utilized by PM4E should provide different levels of diagramming elements to support these different groups. In other words, both information perspective and execution variables, as well as organizations structures should be available in the notation. Moreover, execution requires a notation which would allow producing detailed models while descriptive purposes put more emphasis on understanding and therefore they should allow modeling high level process models. Since PM4E focuses on using one single model for both perspectives, the notation should also enable transformation of elements among different perspectives.

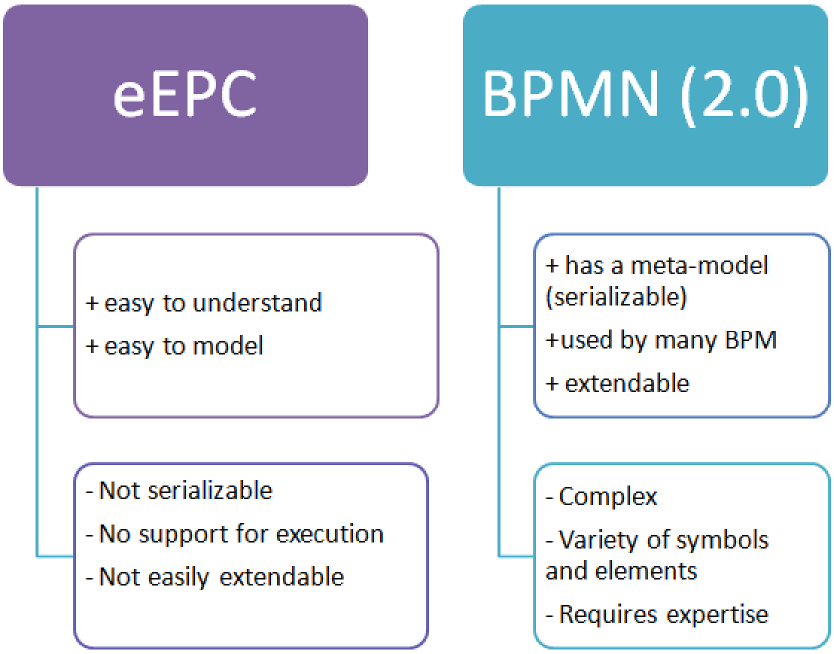
Another important requirement for PM4E is related to process execution. Processes are executed using process engines and they require certain formats to understand the model. Each process engine uses different formats for executable processes. Since process execution is an important aspect of PM4E, the notation to be utilized by PM4E should be transferrable into different formats. This requires a complete meta-model for the notation in order for it to be serializable.

To summarize, the notation for PM4E should have the following properties:

- It should be suitable for producing process models which are high-level and focuses on understanding the processes, mainly to be used by business users.
- It should be suitable for extending the process model with execution variables in a detailed level so that they would be understandable by a process engine.
- It should be extendable since process modeling for execution requires many variables which may not be standard, i.e. changing according to specifications of the process engine.
- It should provide means to transform the notation to other notations.

3.3.2 Usability of Available Notations

When we consider the notations described in Chapter 2, eEPC and BPMN comes forward as commonly used business process modeling notations. When we compare these notations, as depicted in Figure 3.13, we see that eEPC is strong for producing understandable process models. It is preferred by many users since it is simple and easy to understand. This is very suitable for business modeling, ie. Phase I of PM4E. However, eEPC does not provide any means for process execution. Moreover it does not have a complete meta-model, therefore a complete model transformation is not possible with eEPC. On the other hand, BPMN, especially after introduction of BPMN2.0, focuses on process execution, provides many elements for extension. It may be transformed into other notations since it has a complete meta-model. In this sense, it is very suitable to be used in Phase II of PM4E. However, due to the complexity and variety of process symbols and elements, the notation itself is not easily understandable and usually requires deep knowledge and experience.



eEPC

+ easy to understand

+ easy to model

- Not serializable

- No support for execution

- Not easily extendable

BPMN (2.0)

+ has a meta-model (serializable)

+used by many BPM

+ extendable

- Complex

- Variety of symbols and elements

- Requires expertise

Figure 3.13: eEPC versus BPMN

The notation PM4E requires extensive support for process execution which requires considerable degree of effort and extension to be used for business process modeling with eEPC.

BPMN 2.0 is powerful in the sense that it offers valuable enhancements especially for execution:

- Each element has execution semantics
- It provides extensibility opportunities especially with tools implemented with Eclipse
- It provides notation for human interaction

The major advantage of BPMN2.0 over other modeling notations is the fact that it has its defined execution semantics and a meta-model for serialization, which can be seen in ??. This results in being able to store BPMN models, together with technical details, as XML files. Therefore, we have utilized BPMN 2.0 notation, not because it is a superior process modeling notation but because its support for process execution, as the basic notation for PM4E and tried to overcome its disadvantages especially for business process modeling perspective. The major disadvantages PM4E tries to overcome are:

- Separation of concerns between business and executable model
- Providing portability according to specific process engines
- Lacking of graphical interchange mechanisms makes transformed models difficult to understand.

3.3.3 Structure and Elements

As mentioned in the previous section, the basis of the notation relies on BPMN2.0 notation. The symbols, elements and syntactic and semantic rules of BPMN2.0 is utilized in PM4E. However, BPMN 2.0 consists of approximately 360 graphically different symbols. Presenting users with this variety makes the modeling cumbersome for them. In order to overcome the complexity of this notation, especially for business users, PM4E uses conformance class structures of BPMN.

3.3.3.1 Process Diagram with Business Perspective

In this perspective, the focus is on producing simple, understandable and easy to use process models, therefore, we use simple and descriptive conformance subclasses of BPMN2.0. The elements supported by PM4E business perspective are listed in Figure 3.14.

The concept of conformance class is introduced with BPMN 2.0. Although many tools claim to support BPMN the degree of support is important for selecting the tool to diagram the process models.

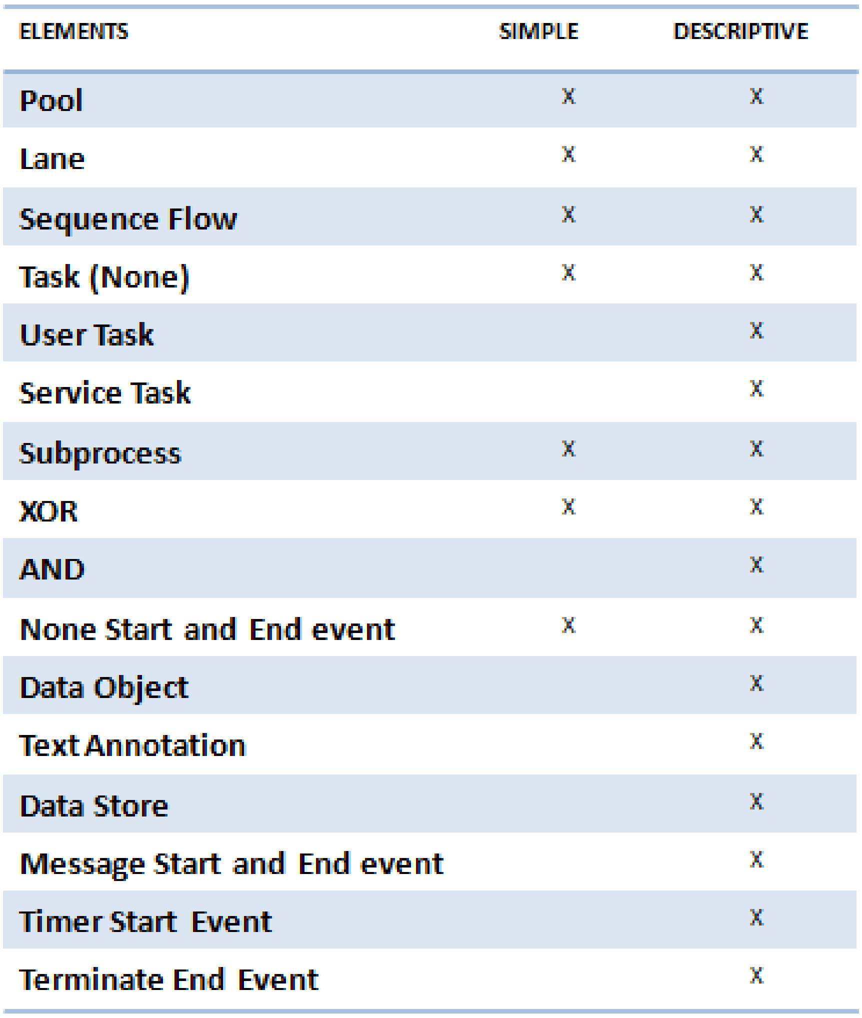
3.3.3.2 Process Diagram with Execution Perspective

In this perspective, the focus is on producing detailed and executable process models, therefore, we use analytical and common executable conformance subclasses of BPMN2.0. The elements supported by PM4E execution perspective are listed in Figure 3.15.

Common executable conformance class also requires to use:

- XML Schema as the definition language
- WSDL for web-service integration
- XPATH for data access language (i.e. parsing BPMN documents for specific elements)

The use of execution related information, such as process variables, data definitions, user interface definitions, in BPMN2.0 notation are explained in section 3.4 where the tool for PM4E is discussed.



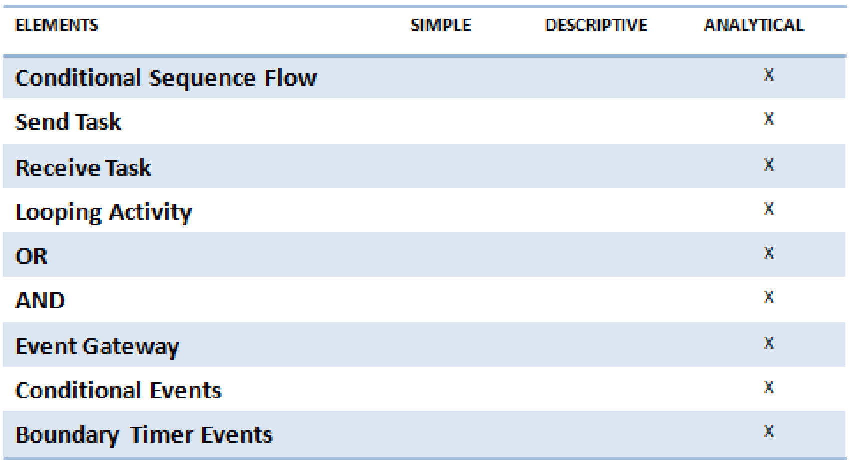
ELEMENTS	SIMPLE	DESCRIPTIVE
Pool	X	X
Lane	X	X
Sequence Flow	X	X
Task (None)	X	X
User Task		X
Service Task		X
Subprocess	X	X
XOR	X	X
AND		X
None Start and End event	X	X
Data Object		X
Text Annotation		X
Data Store		X
Message Start and End event		X
Timer Start Event		X
Terminate End Event		X

Figure 3.14: Elements in Business Perspective

3.3.4 Model Transformations

Although BPMN2.0 provides many elements for executing processes directly, this notation is not utilized by process engine vendors, yet. Most engines use their specific formats. PM4E uses Activiti Server to deploy and execute processes and Activiti requires processes to be modeled using bpmn20.xml format - which is a format meaning fully compatible with BPMN 2.0 specification, whereas many tools provide bpmn.xml for producing the xml representation of the process model. Although, these should be compatible - even if they are not fully compliant - since they both rely on BPMN notations, some issues regarding especially new elements introduced in BPMN 2.0 are problematic.

PM4E alters the model transformation in order to make it executable in Activiti server. The details are described in Section 3.4 where the tool is described.



ELEMENTS	SIMPLE	DESCRIPTIVE	ANALYTICAL
Conditional Sequence Flow			X
Send Task			X
Receive Task			X
Looping Activity			X
OR			X
AND			X
Event Gateway			X
Conditional Events			X
Boundary Timer Events			X

Figure 3.15: Elements in Execution Perspective

3.3.5 Organizational Process Definition Dictionary

PM4E provides guidelines on how to model the processes starting with a descriptive point of view and escalates through execution. In this process, it is helpful to produce a dictionary within the organization in order to standardize the concepts used especially in process execution.There are two major areas which require integration with organizational systems and to maintain consistency , we offer to construct two dictionaries for these areas:

1. Data Types: PM4E guides modelers on how they will include data types in the process model, however, it does not provide any specific data structures since they depend on organization's selected process engine and web-service language. PM4E encourages to construct a data type dictionary within the organization such that modelers can look up which data structure can be used for which elements.

2. Role Definitions: Every organization has its own authorization and authentication systems. In its current state, integrating these systems with BPM systems has a long way to go. However, we propose to use role names in accordance with the ones used in organization's legacy authorization/authentication systems so that common language for integration will be constructed.

3.3.6 Execution

Although BPMN2.0 claims to produce ”directly executable” processes from the diagram, in order for this claim to be true, there must be process engines which would interpret this model and execute accordingly. In its current state, however, process engines does not support directly executable business models. Each engine requires its own notation. For example, Intalio Server requires the processes to be deployed in bpel and wsdl notations, on the other hand, Activiti server requires bpmn20.xml notation to deploy the processes to the server.

In order to execute processes, we have analyzed two open-source process engines; Intalio and Activiti. Intalio engine is built on Apache ODE BPEL engine. Intalio server is a native BPEL 2.0 server which is based on J2EE. Intalio BPM also has a designer for process modeling. If this designer is used for process modeling, necessary BPEL codes for deployment are produced within the modeler and therefore the deployment may be easy. However, if only the Apache server will be used, producing server compliant BPEL and WSDL codes are very difficult. A deployment descriptor including process definition in BPEL format, and service definitions in WSDL format should be produced. This formats does not provide graphical representation, therefore, transforming them requires extensive knowledge of BPMN, BPEL and WSDL.

Activiti is a light-weight process engine for Java supporting BPMN2.0, which is distributed under the Apache license. It also provides a modeler, however, it is easy to transform the processes without using this modeler since it is compliant with BPMN2.0. The main benefits of Activiti are [76]:

- It runs on any Java environment like Spring, JTA, standalone.
- It is easy to get up and running with the setup utility

Since Activiti is easy to setup, fully compliant with BPMN2.0 and requires less effort to deploy processes PM4E uses Activiti as the process engine, and bpmn20.xml as the deployment notation.

3.4 The Tool for PM4E

To be able to demonstrate the usability of PM4E method, it should be supported by tools that are used to produce, maintain and transform business processes. In this section, we firstly describe the high-level requirements for a tool to support PM4E method, and then describe the tool and the extensions performed on the tool.

3.4.1 Tool Requirements

This section discusses the high-level requirements to support PM4E method described in previous sections. Overall, the tool should:

- enable customization and extendibility
- be easy to set up

- provide syntactic and semantic checks for modeling
- enable phase-based process modeling

The requirements for the tool are listed in Table 3.3.

Table 3.3: Requirements for PM4E Tool Support

Requirement	Description
R1 - The tool shall enable process modeling for different perspectives	PM4E offers two perspectives, namely business and execution. The notations used in these perspectives are different and the tool should be customizable so that the different element sets may be applicable to different perspectives.
R2 - The tool shall support semantic rules and list errors, if there are any	
R3 - The tool shall support syntactic rules and list errors, if there are any	
R4 - The tool shall present the process model to users according to selected perspectives	Since there are many symbols in BPMN2.0, only relevant elements should be visible in the selected perspective. The tool should provide a conversion mechanism to display unsupported elements in the business perspective which are present in the execution perspective.
R5 - The tool shall maintain the up-to-date status of model for different perspectives	The changes made from one perspective should be immediately visible to other perspectives.
R6 - The tool shall provide transformation for execution	The process model should be exportable to BPMN20.xml format so that it would be executable within Activiti.
R7 - The tool shall provide data definition interface	
R8 - The tool shall provide user interface definition	
R9 - The tool shall enable extending BPMN2.0	

3.4.2 Usability of Available Tools to Support PM4E

There are many tools for providing business process modeling environments. We have considered four of them to select the tool to expand according to PM4E requirements. We have selected ARIS because it is powerful in business perspective. Intalio Designer, Activiti Modeler and Eclipse Modeler are open-source environments which have extensibility opportunities. We have made the comparison regarding the requirements defined in the previous chapter. The details are listed in Table 3.4

Table 3.4: Comparison of Existing Process Modeling Tools

Requirement	ARIS	Intalio Designer	Activiti Designer	Eclipse BPMN 2 Modeler
R1	No	Yes	No	Yes
R2	No	Yes	No	Yes
R3	No	No	Yes	Yes
R4	No	No	No	No
R5	No	No	No	No
R6	No	No	No	Partially
R7	No	Yes	No	Yes
R8	No	Yes	Yes	No
R9	No	No	Yes	Yes

Intalio and Activiti modelers are especially designed for their process engines. Although they may be used for generic process modeling approaches they are not suitable for customization. Eclipse Modeler is specifically designed to support BPMN2.0 notation and it provides extension points to customize the tool, therefore, we have selected Eclipse BPMN2.0 Modeler as the basis tool to support PM4E.

The next section describes the tool and the extensions performed on the tool to support PM4E method.

3.4.3 Eclipse BPMN2.0 Modeler and Extensions

Considering the requirements for the tool, PM4E would be supported either by the selected product or a separate tool should be developed from scratch. Since BPMN2.0 is a very complex and comprehensive specification, with over 500 pages, we have decided that implementing all the syntactic and semantic controls would not be worth the effort, since the main focus of the tool is demonstrating the usability of the method. We have chosen Eclipse Modeler and developed a prototypical approach to demonstrate that the method brings value to business and IT users.

Eclipse Modeler is an open-source SOA project aiming to provide basis for building extensible tools to enable the design, configuration, assembly, deployment, monitoring, and management of software designed around a Service Oriented Architecture [77]. The project is guided by the values of transparency, extensibility, vendor neutrality, community collaboration, agile development, and standards-based innovation [77].

The increasing amount of interest for BPMN2.0 notation resulted in the launch of this project. The BPMN2 Modeler is a graphical modeling tool which:

- is based on Eclipse Graphiti. Graphiti is a graphical tooling infrastructure which provides a modeling environment focusing on graphical representations and customization opportunities.
- supports producing BPMN diagrams using BPMN2.0 EMF meta-model. This meta-model is being developed within the Eclipse Model Development Tools (MDT) project [77].
- is compatible with the BPMN 2.0 specification proposed by the Object Management Group, however, it does not support all the elements, it provides a reference for the modeling notations.
- enables building various EMF-based modeling tools.

We have used Eclipse 3.7 (Indigo) as the development environment. In order to set up the project to run, we have downloaded the projects listed in Table 3.5 from Eclipse download site. The major projects, which were edited to support PM4E requirements are described as follows:

- org.eclipse.bpmn2.editor: This project provides the details for BPMN2 editor.
- org.eclipse.bpmn2.modeler.ui: This projects defines the implementation of the user interface of the BPMN2 editor.

The changes implemented in the projects regarding the requirements may be listed as follows:

- R1 - The tool shall enable process modeling for different perspectives: The package named org.eclipse.bpmn2.editor is extended with extension points to include perspectives utilizing ”org.eclipse.ui.perspectives” implementing IPerspectiveFactory.
- R2 -The tool shall support semantic rules and list errors, if there are any: Eclipse BPMN2 Modeler supports the semantic rules of supported BPMN elements. The errors are represented graphically to modelers, i.e. the elements has an information or error box on top left corner of the element. The graphical display of error or warnings may be confusing for the business users, however, for execution steps they are beneficial.
- R3 - The tool shall support syntactic rules and list errors, if there are any: Eclipse BPMN2 Modeler supports the syntactic rules of supported BPMN elements. The errors

Table 3.5: Eclipse Projects to Run Eclipse BPMN2 Modeler

Project	Github Project
org.eclipse.bpmn2	bpmn2
org.eclipse.bpmn2.edit	bpmn2
org.eclipse.bpmn2.editor	bpmn2
org.eclipse.bpmn2.feature	bpmn2
org.eclipse.bpmn2.modeler.core	org.eclipse.bpmn2-modeler 0.2.0-Final
org.eclipse.bpmn2.modeler.examples.feature	org.eclipse.bpmn2-modeler 0.2.0-Final
org.eclipse.bpmn2.modeler.feature	org.eclipse.bpmn2-modeler 0.2.0-Final
org.eclipse.bpmn2.modeler.jboss.runtime.feature	org.eclipse.bpmn2-modeler 0.2.0-Final
org.eclipse.bpmn2.modeler.runtime.example	org.eclipse.bpmn2-modeler 0.2.0-Final
org.eclipse.bpmn2.modeler.runtime.jboss.jbpm5	org.eclipse.bpmn2-modeler 0.2.0-Final
org.eclipse.bpmn2.modeler.runtime.jboss.jbpm5.tests	org.eclipse.bpmn2-modeler 0.2.0-Final
org.eclipse.bpmn2.modeler.ui	org.eclipse.bpmn2-modeler 0.2.0-Final
org.eclipse.bpmn2.modeler.updatesite.feature	org.eclipse.bpmn2-modeler 0.2.0-Final
org.eclipse.bpmn2.tests	bpmn2
org.eclipse.bpmn2.tools.ecoremerger	bpmn2
org.eclipse.bpmn2.xsltFromEcore	bpmn2

are represented graphically to modelers, i.e. the elements has an information or error box on top left corner of the element. The graphical display of error or warnings may be confusing for the business users, however, for execution steps they are beneficial.

- R4 - The tool shall present the process model to users according to selected perspectives: org.eclipse.bpmn2.modeler.ui is extended to include two user interfaces for each perspective in order to restrict the number of elements used for each modeling phase.
- R5 - The tool shall maintain the up-to-date status of model for different perspectives: PM4E focuses on producing one single model for each perspective with the ability to present users with relevant elements, hiding the unwanted details. This is especially a concern for business users, since all the elements in the business model exists in the execution model however, all the elements in the execution model does not exist in the business model. In order to present the process model to business users with a restricted set, a mapping from execution conformance class to descriptive is required. The following rules apply in this conversion:
  - Conditional sequence flow is displayed as a sequence flow.
  - Send and receive tasks are displayed as none tasks.
  - Looping activity is treated as a none task.
  - OR and AND is converted to XOR and AND.
  - Event gateway, conditional events and timer events is displayed as none events.
- R6 - The tool shall provide transformation for execution: The following extensions are done in the bpmn model.

  - Process tag is extended with isExecutable attribute.
  - For each element roles are defined. We define the role such as activiti:assignee= "${ProgramManagerExpert}"
  - For each element inputs and outputs are defined. We define these data using the globally defined iospecifications and adding these to the model using the format activiti:dataName="${DataName}"

- R7 - The tool shall provide data definition interface: Eclipse BPMN2 Modeler provides a user interface to support input and output definitions using BPMN2.0 iospecification tags. We use this syntax to transform this definitions to Activiti bpmn20 format as explained in previous requirement.

- R8 - The tool shall provide user interface definition: Eclipse BPMN2 Modeler does not provide a user interface for user interface definitions. We extend the model using the transformations required by Activiti engine in order to include user interface details. We fulfill this requirement manually currently, extending BPMN2 modeler for user interface design is left as a future work. For each element which requires human intervention we extend the model with extensionElements extension point using formProperty tags.Whe the forms are defined in bpmn2.0 xmp format, Activiti engine uses its FormService to create and submit user forms in the run time.

- R9 - The tool shall enable extending BPMN2.0: Since Eclipse BPMN2 Modeler is an open-source application, the tool allows extending the produced models.

CHAPTER 4

APPLICATION OF PM4E

This chapter presents the application of PM4E method in a multiple-case study with three case studies; first in software development division of a governmental organization. Other cases were implemented in another governmental organization which initiated a project to define its business processes and develop information systems to execute these processes. In this chapter, first section describes the design of our multiple-case study elaborating on the questions of the study, the propositions, and data collection and analysis strategies. In the following sections, we describe the conduct of the case studies and discuss our findings.

4.1 Multiple Case Study Design

The initial step of this study started with a comprehensive literature review to identify the gaps and improvement opportunities for using descriptive process models for execution. This review provided us the fact that process execution has a very wide span of different disciplines and guided us on defining the span of this study.

Case studies are frequently used common research strategies in many disciplines such as political science, psychology, sociology, anthropology, business, social work and information systems [78], [79], [80], [81],[82]. They help researchers to understand and analyze the problem in its context [83]. Case study strategies are very beneficial for capturing the knowledge and experience of investigators in order to develop generalizations. They allow researchers to answer ”how” and ”why” questions to understand the nature and the complexity of the operations [84]. They rely on multiple sources of evidence and benefit from the prior development of theoretical propositions[85].

Case studies are best suited to considering the how and why questions, or when the researcher has little control over events [81]. Also, case studies are especially well suited to new research areas or research areas for which existing theory seems inadequate[85]. Both of these conditions are valid for our situation, therefore, we preferred the case study as an appropriate research strategy to investigate the application of the PM4E method in real-life organizational settings, to collect data to answer the research questions and to examine its implications.

Our research has both exploratory and explanatory applications. We initially conducted an exploratory approach to identify the current business process modeling and execution modeling domains to identify the status of the current modelling tools and methodologies in the modeling environment. This study identified the gaps between this domains and offered the improvement opportunities. In order to explore the applicability of our method for utilizing descriptive process models for execution and explore possible improvement opportunities, we conducted a multiple-case study involving three cases. First case study was conducted in a software development division of a governmental organization to identify the problem and explore the solution approach. Second and third case studies were conducted to show the applicability of the method in another governmental institution and third case study was implemented to show the extendibility of the method. According to Yin [81], there are six possible sources of evidence for case studies: documents, archival records, interviews, direct observation, participant-observation and physical artifacts. We have used four of them which are documents, direct observation, participant-observation and physical artifacts.

Case studies can be designed in many ways; but the primary distinction in designing case studies is between single- and multiple-case designs [81]. Single case designs are generally for instances when there are no other cases available for replication. Multiple case designs include single cases, but design must follow a replication rather than sampling logic for each single case. In this thesis study, we used multiple-case design strategy, since; a multiple case is often considered more compelling as it strengthens the results by replicating the pattern matching and increasing confidence in the robustness of the theory. The design of the case study used in this thesis is depicted in Figure 4.1.

We have developed the PM4E methodology starting with an extensive literature review. After developing the methodology, as the figure illustrates, we have defined case selection criteria and formed data collection design. After selecting the cases, we have conducted each case study and analyzed the results. These analysis provided inputs for improving the methodology. Especially, the first case study was very productive in terms of identifying the gaps between business models and execution models. As the last step, we analyze the case studies in relation to our research questions and propositions.

4.1.1 Case study questions and propositions

Considering the different focuses of the case studies, our research questions and related propositions are described in terms of the cases.

Our first case study was mostly an exploratory one focusing on the requirements of a methodology considering executability components. In this case study, we have explored the requirements for the PM4E method. Furthermore, basing on the findings of this case, the method and other components of the PM4E were improved. The research questions for this case study were as follows:

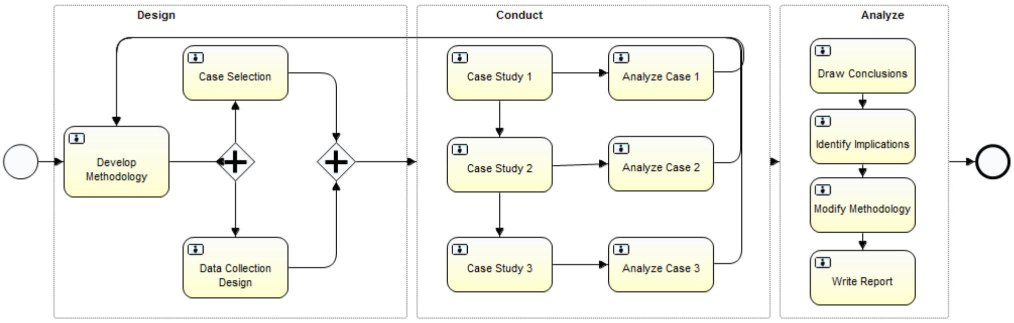


Figure 4.1: Case Study Design

- Question 1: What are similarities and differences of business models and execution models? Business models and execution models are developed by modelers with different focuses therefore the abstraction levels of these models are different. Not only the model abstraction, but also the notations and technologies used for process execution is different from business modeling. In this case study, we identify the similarities of two models as well as the key requirements to execute a process.
- Question 2: How can business process models be used for process execution?
  - Proposition 1: We can use a process modeling method so that we can utilize business process models for process execution .

The emphasis in the other cases was on exploring and evaluating the applicability of the method and observing expected benefits. Our multiple-case study has the following primary research question:

- How applicable is the PM4E method for using descriptive process models for process execution?

  We propose that the method PM4E is applicable for modeling organizations' processes in a way that can be used for execution and with this method all stakeholders in the modeling domain benefit, increasing the productivity of the organization.

  - Proposition 2: The method helps maintaining a common understanding for IT and business users. Process models can be described in various abstraction levels. Business users prefer high-level and understandable models for description, however, IT users need comprehensive, detailed and complex process definitions for execution. Although both type of models have commonalities since they describe the same process, the output models for these perspectives differ significantly. PM4E method provides a modeling methodology for both perspectives on the same model so that each perspective is presented without the overwhelming details not required by that perspective.

- Proposition 3: An organization following the PM4E method for modeling its processes can perform process modeling in less time. PM4E proposes that business modeling and execution modeling should be conducted with different element set which results in isolating irrelevant and confusing elements, especially for business users. In other words, guiding the users in the process of modeling while blocking the complexity of process modeling environment, reduces the total effort. In order to validate this statement, we propose comparing the results of modeling processes by different - but with similar qualifications - users with different modeling tools.
- Proposition 4: There are various stakeholders responsible for escalating from descriptive business models to executable models. All the requirements regarding these stakeholders should be present in the model to be able to execute it. An executable process model - in which all the requirements are fulfilled - reflect organization's final and updated processes. Related researches state that process models which are formed by including all relevant participant are more successful in terms of their completeness [86], [87], [88], [89]. PM4E integrates business users, who have process knowledge, and IT experts, who have execution knowledge, therefore we can easily claim that PM4E includes relevant stakeholders in process modeling process. Furthermore, since PM4E encourages including managers from business level as well as process owners, we can claim that PM4E will produce complete process models.
- Proposition 5: One single process model will guide different interest groups from modeling to execution; therefore, process will maintain its dynamism and up-to-date status.

4.1.2 Selection of the cases

Although case studies are extensively used they have their own limitations, and misapplication can produce incorrect or inconsistent findings. Therefore, it is very important to define the formal mechanisms of the research methodology design to overcome the pitfalls. One important issue for the design is the selection process. To be able to answer the research questions, the selected case should comply with the following criteria:

- To be able to demonstrate the executability of the processes, the processes should contain complex elements such as loops, inputs and outputs, decision points, user interactions and business rules. These attributes assure that the proposed solution will be applicable to not only a limited type of processes but may be usable for various type of processes from different domains as well. Processes selected with these criteria will enhance the idea that the results will be generalizable and applicable for other processes intended to be executed.

- To ensure that the business and executable models are always up-to-date reflecting the current status of how the task is actually performed, the processes should have related subprocesses. This way, we will be able to analyze how a change performed in a suprocesses will be reflected to the overall description and execution of the overall process.

- Since the processes will actually be executed, interactions with other organizational systems and processes should be kept at a minimum. Each organization may have its proprietary means of conducting its work, which is incorporated in process execution environment by means of web services. Integration with organizational systems has considerations beyond process modeling escalating through enterprise modeling, which is not in the scope of this study.

Ideally, the processes should be selected in an organization having a process execution environment of its own. In this way, using PM4E methodology we would have compared the results to their execution methods. However, organizations with process execution initiations are very rare, therefore, we have selected processes from two organizations executing them with traditional software development methodologies, and compared the results.

First case study differs from the second and third one in the sense that their aims are different. Therefore, we need to plan and conduct the study for these case differently. In the first case, we focus on exploring the gaps between descriptive and execution models, in the following ones we apply the methodology and find improvement opportunities.

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 separately in their sections. For the multiple case study, the following general plan is followed:

- Conduct Case Study 1 to identify the similarities and differences among process models used by business users and IT users. The focus is on forming the methodology and identifying improvements.

- Conduct Case Study 2 and 3 to verify that PM4E can be used effectively for utilizing decriptive process models for process execution.

4.2 Case Study 1 : Change Management

This case study was conducted to explore the requirements of a unified modeling approach which would combine business and IT perspectives together in itself.

4.2.1 Background

The first case study is conducted in the software development unit of a governmental organization. The organization in question was launching a process improvement initiation and Change Management Process was selected as a test case. In the change management process, users from various units (business units, help desk, software development team) report problems concerning software in production or request new functionality, which is expresses as a change request. At first step, these requests are analyzed by the development team in terms of its feasibility. If the change request is accepted, a deeper analysis is carried and cost estimation is conducted. Taking manager's opinions into consideration they decide whether to implement the changes and when and by whom.

The definition for the change management process was produced in natural language. Because of this definition, there were many ambiguities and it was very difficult to comprehend. When we analyzed the process, we have seen that it was a very suitable and challenging opportunity to produce descriptive models and escalate through executable model definitions. Furthermore, the nature of the process was very suitable for execution since user interfaces were at a minimum level.

4.2.2 Case Study Plan

In this case study, we have modeled the processes looking from the business users' perspective, and then, we investigate the usability of these models for process execution efforts, rather than modeling the processes starting from scratch. The objective of this case study is identifying the degree of executability of business process models and defining the requirements of process execution

The steps for each of the case studies are as follows:

1. Develop business process models in eEPC notation.

2. Select process execution environment. The must-have component in order to execute process models is the process engine. Although there are several free process engines available, each one of them uses different notations and technology. It is very important to identify which engine to use, since it will guide the modelers to define execution variables according to the specifications of the engine.

3. Identify required inputs for execution engine.

4. Update business model so that it will be executable with the selected engine

5. Compare efforts for modeling and execution and discuss the results.

4.2.3 Sources of Evidences

In this case study we have used the following sources of evidences, as offered by Yin [81], to analyze and evaluate the results.

- Documentation: In this case study, we use existing process definitions expresses in a natural language to understand the as-is process.
- Physical Artifacts: We have executed the process models and therefore produced the following outputs:
  - Business process models, with EPC and BPMN notation
  - Interface and Data Definitions
  - Executable process models in BPEL and WSDL formats
  - Effort Data in terms of man-hours
- Participant Observation: In this study, the models are evaluated by the people who defined the as-is processes.

4.2.4 Case Study Conduct

This case study was conducted to explore the possibility of using business process models for execution purposes. We have used eEPC and BPMN for modeling the process and utilized Intalio Business Process Management Suite to execute them.

4.2.4.1 Premodeling

The change management process was originally defined by a group of people selected from each department of IT unit in the organization. These people were given the responsibility of consulting with relevant project groups and collecting their requirements. When the first draft of the definition was ready, it was shared by all relevant participants and the feedbacks were collected. It became clear that some project groups were not satisfied with the produced definition since their requirements were not met. The defined process took much time and therefore it was affecting their efficiency.

The major objective while conducting this case study was exploring the execution opportunities, assuming that once there was a process definition it could be modeled in synchronization with all participants. But once we have started discussing the definition with people from different units and project groups in the organization, it became clear that having a defined process model does not ensure that this process is accepted in the organization.

During this case study we have decided to add this process analysis phase to ensure that each stakeholder has the same understanding and every requirement of each group is met.

4.2.4.2 Business Modeling

Change management process is modeled with eEPC notation for an explicit definition. High level model can be seen in Figure 4.2, detailed models are listed in Appendix A. In this phase, the roles, organizational units and input/output relations are made clear. Although, there are some tools supporting transformation among various notations, at the time of this study they were not mature enough. Main reason was due to fact that BPMN did not have a structured meta-model supporting serializability. Therefore, the processes modeled with eEPC could not be directly used in any execution environment. In order them to be executable, we had to remodel them in an environment supported by the execution engine, or transform them into such an environment.

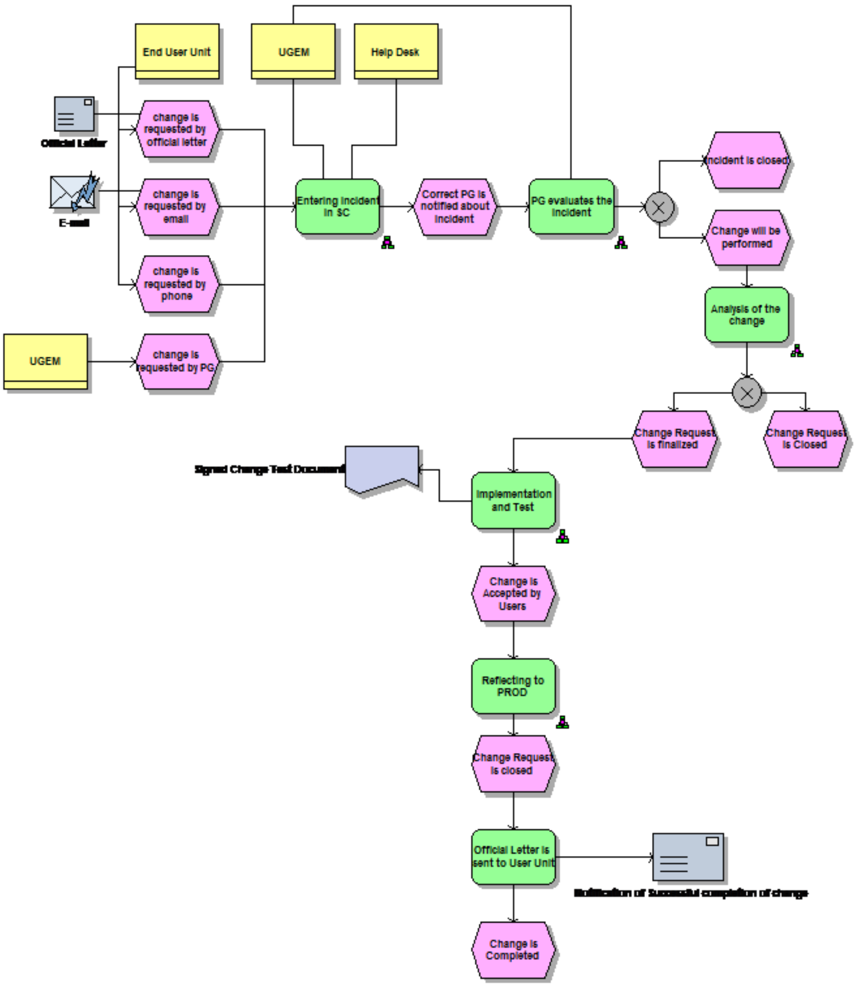

End User Unit
UGEM
Help Desk
Official Letter
change is requested by official letter
E-mail
change is requested by email
change is requested by phone
UGEM
change is requested by PG
Entering incident in SC
Correct PG is notified about incident
PG evaluates the incident
incident is closed
Change will be performed
Analysis of the change
Change Request is finalized
Change Request is Closed
Signed Change Test Document
Implementation and Test
Change is Accepted by Users
Reflecting to PROD
Change Request is closed
Official Letter is sent to User Unit
Notification of Successful completion of change
Change is Completed

Figure 4.2: Business Model with eEPC Notation

In this case study, we have remodeled them with BPMN to identify the effort, using Intalio Designer. The PM4E business modeling phase is conducted as follows:

- Identify process goals: The main objective in this process is answering business users' change request by either implementing or rejecting it.

- Identify roles: The following roles are defined to conduct this process:
  - Business unit starts the process by requesting a change in a current software. They also have the responsibility of testing the changes made by IT.
  - Business unit manager approves the completed change so that it can be reflected to production environment.
  - Help desk makes the project group aware of the request and the business unit of the current status of the request.
  - IT project manager is responsible with analyzing the request, deciding on the size, preparing the schedule and assigning the change to an IT developer.
  - IT developer implements the required changes.
  - IT manager decides if the required change should be implemented basing on the evaluation provided by the project manager.
  - Deployment responsible reflects the changes to production environment.
  - Service center is the application used by help desk to manage the coordination between various units.

Table 4.1: Case 1 Activity Definitions

Activity	Role
Create new incident	Help Desk
Assign Incident to related project group	Help Desk
Find correct project group	Help Desk
Obtain initial requirements	Project Group
Reject request	Project Group
Complete the impact analysis of change	Project Group
Estimate the Cost of Change	Project Group
Assign Developers to tasks	Project Manager
Implement	Developer
Test	Business User
Approve the change	Business Manager
Reflect changes to PROD	Deployment Responsible
Send Notification to help desk	Service Center
Close the Change	Help Desk

- Major activities in the process are defined in Table 4.1, in the actual sequence of the process. In this step we are only interested in the major activities and who carries them. We also identify the start and end events in the process.
  - Start: Process is started when a request is submitted from business unit via email, help desk, phone or an official letter, forming the change request.

  - End: Process may end in three ways. First one is realizing the change and reflecting it to production environment. Second ending condition occurs after the initial evaluation of the process if it is considered to be infeasible, therefore the change request is not realized. The final ending condition may occur if an IT manager declines the request due to the cost estimation report prepared by the project manager.

- Identify major inputs and outputs: At this point we have the major activities with the roles who perform them. In this step we identify the input output relations, as depicted in Table 4.2.

Table 4.2: Case 1 Activity Definitions with Input and Outputs

Activity	Role	Input	Output
Create new incident	Help Desk	Official Letter or Email	
Assign Incident to related project group	Help Desk		Project Group Notification
Find correct project group	Help Desk		
Obtain initial requirements	Project Group		Feasibility Report
Reject request	Project Group		Notification to Business Unit
Complete the Impact Analysis of change	Project Group		Impact Analysis Report
Estimate the Cost of Change	Project Group	Impact Analysis Report	Cost Estimation Report
Assign Developers to tasks	Project Manager		
Implement	Developer	Impact Analysis Report	
Test	Business User		Signed Test Document
Approve the change	Business Manager		
Reflect changes to PROD	Change Responsible		
Send Notification to help desk	Service Center		
Close the Change	Help Desk		Notification to Business Unit

- Define conditions: In this process, the error cases mentioned by participants were mostly related to the service center application used by help desk, such as not finding the project group information, and deployment operations employed by deployment responsible. These applications were out of the scope of this process therefore they are not included in the process definition. In this step we explore specific conditions for each activity to be performed as depicted in ref{table:case1condition

Table 4.3: Case 1 Condition Definitions

Activity	Condition
Create new incident	
Assign Incident to related project group	
Find correct project group	
Obtain initial requirements	
Reject request	
Complete the impact analysis of change	
Estimate the Cost of Change	If cost estimation is greater than 5 man-days consider manager's approval
Assign Developers to tasks	
Implement	
Test	
Approve the change	
Reflect changes to PROD	
Send Notification to help desk	
Close the Change	

- Review Process: The process is reviewed by all participants in order to ensure that all roles in the process analysis step exists in the process model, all activities are connected with roles and all input and outputs are associated with the activities.

Once all the required information is gathered, we model the process diagram. In Intalio there was no way to input the eEPC model therefore we have remodeled it using BPMN notation provided by Intalio Designer. The modeling steps are:

- We place the pools to depict who is actually performing the activity.

- We have a list of activities as listed in Table 4.1. Initially, we place the start event. Then, we place these activities in corresponding pools using the Task element. We model end possible events after the activity which results in this ending condition. The next step is to decide on decision points. We model the decisions using gateway notation. Conditions provided in Table 4.3 are helpful in deciding where to use decision points.

- The next step is to associate tasks, gateways and start and events using sequence and message flows.

- The final step is associating the input and outputs defined in Table 4.2 with relevant tasks using data objects.

At the end of this phase, we have the following business model, depicted in Figure 4.3, in BPMN notation.

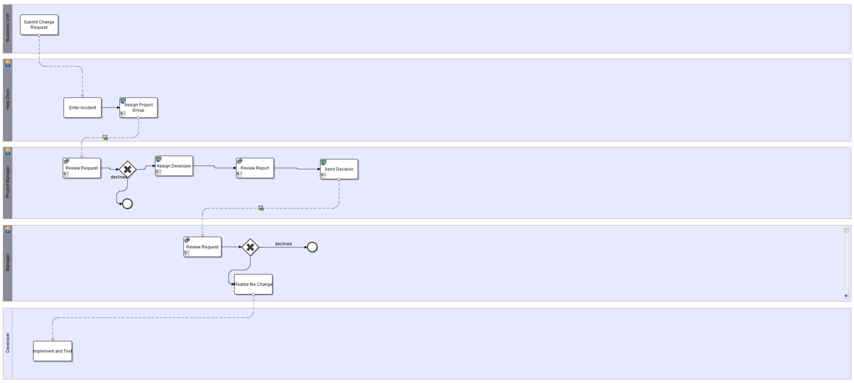


Figure 4.3: Business Model with BPMN Notation

4.2.4.3 Execution Modeling

The methodology for execution modeling raised from the experiences in this step, since the main objective to explore the execution capabilities and the requirements. The following steps are conducted, although not in this order due to structure of Intalio Designer.

- Preparation: As explained in Chapter 3, process execution requires an integrated environment including many components. We have selected Intalio Designer to execute the processes, therefore the requirements for execution are fulfilled according to this suite's appropriate languages and notations.

- User Interfaces: Intalio Designer provides a workflow form to identify the user interfaces utilizing XForms. In this case study we have used this forms to produce the user interfaces. We will use the estimation activity to give the detailed descriptions of user interface design and modeling, as depicted in Figure 4.4.

  In this screen, required elements from the workflow form editor palette is selected and placed in the form. Each element defines an input, output or both to another activity in the process. When the screen is designed, information regarding each item is defined.

- Data Definitions: Intalio Designer enables input and output definitions within forms and within activities. For user interfaces, each element in the interface their behaviors, default values and validation constraints are defined. In estimation screen, there is an

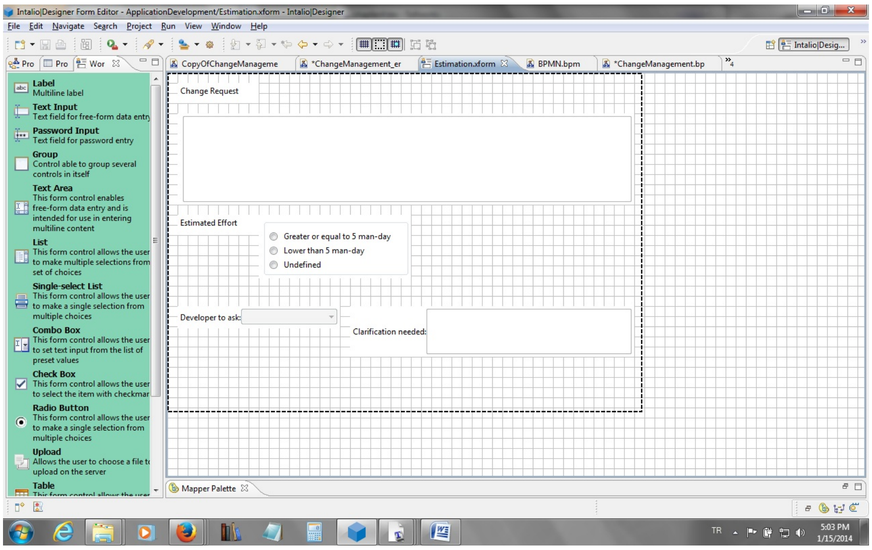

Intalio|Designer Form Editor - ApplicationDevelopment/Estimation.xform - Intalio|Designer
File Edit Navigate Search Project Run View Window Help
Pro
Pro
Wor
CopyOfChangeManageme
*ChangeManagement_er
Estimation.xform
BPMN.bpm
*ChangeManagement.bp
Intalio|Desig...
Label
Multiline label
Text Input
Text field for free-form data entry
Password Input
Text field for password entry
Group
Control able to group several controls in itself
Text Area
This form control enables free-form data entry and is intended for use in entering multiline content
List
This form control allows the user to make multiple selections from set of choices
Single-select List
This form control allows the user to make a single selection from multiple choices
Combo Box
This form control allows the user to set text input from the list of preset values
Check Box
This form control allows the user to select the item with checkmar
Radio Button
This form control allows the user to make a single selection from multiple choices
Upload
Allows the user to choose a file to upload on the server
Table
Change Request
Estimated Effort
○ Greater or equal to 5 man-day
○ Lower than 5 man-day
○ Undefined
Developer to ask:
Clarification needed:
Mapper Palette
TR
5:03 PM
1/15/2014

Figure 4.4: User Interface Definition

input element giving the definition of the screen and three output elements identifying the effort, consulting to another developer and the issues becomes defined.

Figure 4.5: Data Definition

- Model Extension: In this step, the business model is extended to make it executable. Below are some of the changes done in the business model to fulfill the execution requirements.
  - Intalio requires an additional role definition for expressing the ”system to be designed”. Therefore, we extended the model with an additional pool depicting the execution system. In other words we add an extra role defining what the process engine will do. This pool should be set executable whereas other roles should be set non-executable.
  - Events: Change the start event to a message start event since the process begins with a message from the help desk.
  - Assign user interfaces to tasks.
  - For each task, identify response types if the activity requires a response, this is actually the way Intalio defines message types.
  - We detail all the activities such that each activity will depict a single and atomic action.

After finalizing activities, events, messages and gateways, we should define data transformations using Data Mapper, as depicted in Figure **??**, in order to connect input and outputs to process data defined in data definition step..

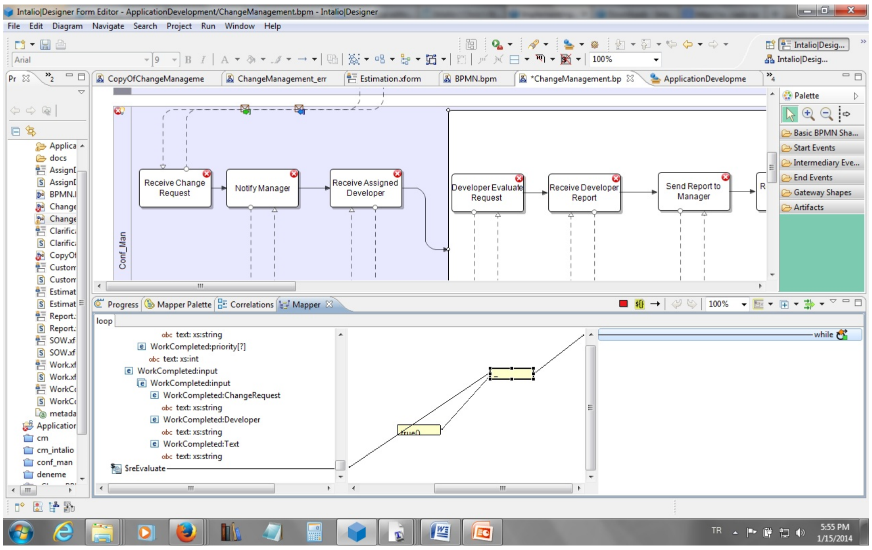


Figure 4.6: Data Mapping

- Business Rules: We have used gateways to implement business rules, since Intalio does not support business rule task.
- Role Integration: We have managed user definitions in the Intalio Designer and Tomcat server's user definitions.
- Execution: In this final step we have an executable process mode as depicted in Figure 4.7, which should be deployed to a server.

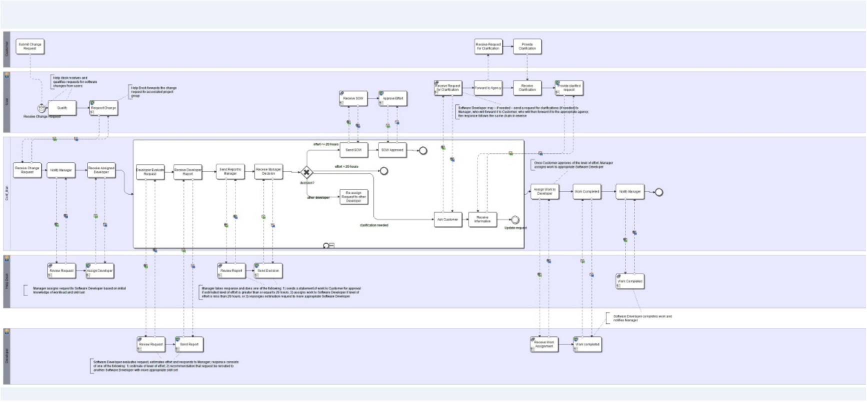


Figure 4.7: Executable Process Model

Intalio mainly provides a deployment wizard which produces the BPEL code for the process, WSDL definitions to represent external services and descriptors for its internal server descriptions. In order to deploy processes, Intalio BPMS server should be up and running. The samples of produced pbel and wsdl deployment files are listed in Appendix A

4.2.5 Analysis

Modeling business processes is a common task for process improvement and process execution and this study explored the extent of reusing process models rather than starting from scratch. It is revealed that the main users for execution engines are those with technical knowledge since it is very difficult for business experts to understand the frameworks. However, the owners of the processes are business people; therefore, the process execution must be incorporated into a more business-logic perspective so that the owners can have more control over their processes.

From a business perspective, business models are well suited for high-level modeling and can include organizational issues such as its goals and structure. On the other hand, execution models do not focus on the business context but rather only support the processes, the data definitions and the interactions between processes including more implementation level details. In order to transform a business model to an executable model, the question ”who will execute what, when, where, how?” must be answered.

This case study provides valuable input for the requirements of process modeling for execution, as summarized in Table 4.4.

We can answer research questions in the following way:

- Question 1: What are similarities and differences of business models and execution models? Table 4.4 reveals the key requirements of execution models. Business models should be extended to include these aspects to overcome the differencies among the objectives of different perspectives.
- Question 2: Can business process models be used for process execution? We can conclude that business models can be used for process execution. However, we should keep in mind that there are many component in process execution domain and not all of them can be fully automated.

4.3 Case Study 2: Updating Performance Management System

This case study was conducted to validate the applicability of PM4E methodology.

4.3.1 Background

This case study uses the processes of a governmental organization which was engaged in a project to define its business processes and develop information systems to support those processes. The organization is a recently established governmental organization aiming to support regional development. One of the main practices of the organization is to develop

Table 4.4: Questions to be answered for process execution

Question	Importance
Who	Those taking part in activities defined in the processes are expressed with enhanced role definitions. However, activity flows between roles may be restricted in business and execution models . In order to increase the approximation between these role definitions , lane-structured role definitions may be utilized to define roles for each activity together with controls for role completeness to assure interactions between roles will be not be problematic in execution models.
What	Some activities in a process may require integration with other web services as defined by the organization. These types of interactions are difficult to identify during the modeling phase and may require an additional step. Moreover, in its current state, an business model lacks execution details since they are not intended for that purpose.
When	The timing of activity in the process life cycle depends on the synchronization managed by web server; when an activity is completed, the proceeding ones are activated. Business models must be enriched with notations to enable the web server understand the orchestration. Moreover, the completion criteria for activities and the following action must be explicitly set.
Where	Web service is the dominant technology for executing processes. However, there are no standards for the implementation of web services; some suites use standards such as BPEL, others use proprietary solutions. Since web services are crucial to the execution, business models should be enriched with support for web service notations.
How	The way in which an activity is performed is defined by inputs, outputs and other web services. Forms are used to define input/outputs between activities; also other tools used by the organizations (email, messaging systems) can be integrated. To define input/output relations business models should incorporate systematic and formal means, which should also allow application and platform specific aspects to be considered.

and conduct regional grant programs. We have selected Updating Performance Management System Process for this case study. The main objective of this process was to produce an annual performance report of the development agencies supported by the organization. Annually, measurement metrics and methods are reviewed, necessary updating are done in the system. Later, data is gathered according to these metrics and after these data is consolidated and analyzed, the final performance report is produced.

The process was originally defined in eEPC notation. We have remodeled the process with BPMN2.0 notation to benefit from the extended attributes of Eclipse BPMN2.0 Modeler defined in Chapter 3.

4.3.2 Case Study Plan

The planned list of activities for this case study is listed below.

1. Develop business model with PM4E methodology
   - Input: Description of as-is processes
   - Data to be collected: Metrics on process details such as number of tasks, gateways and events
   - Data collection method: Metric collection
2. Develop executable model with PM4E methodology
   - Input: Business process model
   - Data to be collected: Analysis of neglected information from business models, number of additional process elements
   - Data collection method: Direct observation, metric collection
3. Produce executable outputs for the specified process engine
   - Input: Executable process model
   - Data to be collected: Time required for transformations
   - Data collection method: Metric collection

4.3.3 Sources of Evidences

In this case study we have used the following sources of evidences, to analyze and evaluate the results.

- Documentation: In this case study, we use existing process definitions expresses in EPC notation to understand the as-is process.
- Physical Artifacts: We have executed the process models and therefore produced the following outputs:
  - Business process models, with EPC and BPMN notation
  - Interface and Data Definitions
  - Executable process models in BPMN20 format
  - Effort Data in terms of man-hours
- Direct Observation: We use direct observation to evaluate the process elements which are discarded in the executable process level.

4.3.4 Case Study Conduct

In this case study, we have used Eclipse BPMN2.0 Modeler for modeling the process and utilized Activiti Business Process Management Platform to execute them. Our objective was to prove the applicability of the PM4E method and identify improvement opportunities.

4.3.4.1 Premodeling

In this step the following base level decisions are agreed on:

- Start of process: Performance Management System is updated based on a schedule, which is yearly.
- End of the Process: The process ends with declaring the performance analysis results.
- Outputs: This process produces an agency performance analysis report.

In the next section the modeling phase following PM4E guidelines is explained.

4.3.4.2 Business Modeling

This process is modeled with BPMN2.0 using the elements in the Business View of extended Eclipse BPMN2 Modeler. In this phase the roles, organizational units, input and output relations are defined and then these are reflected to a BPMN diagram. The PM4E business modeling phase is conducted as follows:

- Identify process goals: Each year the performances of the development agencies are measured and a report is produced. The objective of this process is producing the performance management report by making necessary updates in the performance management system, in terms of metrics and methods, if required.
- Identify roles: There are three major participants in this process:
  - Development agency provides necessary data for performance management.
  - Planning and programming expert is responsible by updating the system, conducting studies for measurement methodologies and analyzing the performance metrics.
  - Program management expert is responsible for gathering and storing performance metrics.
- Major activities in the process are defined in Table 4.5. In this step the major activities together with who carries them are identified. We also define the start and end events in the process.

- Start: Process is started when the time for producing performance management report has come.
- End: Process ends only in one way, which is producing the performance management report.

Table 4.5: Case 2 Activity Definitions

Activity	Role
Review performance goals	PPE
Identify goal questions	PPE
Identify metrics	PPE
Identify methods	PPE
Analyze Infrastructure	DA
Define analysis methods	PPE
Prepare questionnaire	DA
Update infrastructure	DA
Conduct questionnaire	DA
Gather data	DA
Analyze data	PPE
Evaluate data	PPE
Consolidate metrics	PME
Announce report	PPE

- Identify major inputs and outputs: At this point we have the major activities with the roles who perform them. In this step we identify the input output relations, as depicted in 4.6.

Table 4.6: Case 2 Activity Definitions with Input and Outputs

Activity	Input	Output
Review performance goals	Regional Data	Performance Goals
Identify goal questions		Performance Questions
Identify metrics		Performance Metrics
Identify methods	Goals, Questions and Metrics	Analysis Methods
Analyze Infrastructure	Requirements	
Define analysis methods		
Prepare questionnaire		
Update infrastructure	Requirements	
Conduct questionnaire		Questionnaire Data
Gather data	Questionnaire and External Data	
Analyze data		Consolidated Data
Evaluate data		Performance Report
Consolidate metrics		Agency Grade
Announce report	Agency Grade and Performance Report	Performance Report

- Define conditions: No error conditions are required in this process.
- Review Process: The process is reviewed by all participants in order to ensure that all roles in the process analysis step exists in the process model, all activities are connected with roles and all input and outputs are associated with the activities.

Once all the required information is gathered, we have modeled the process diagram using Eclipse BPMN2.0 Modeler.

- Lanes depicting different roles are diagrammed.
- Using the activity list, we place each activity in corresponding lane. We use UserTask notation since these process requires user operations. After the tasks are modeled, we define gateways. In this process there are some activities performed synchronously, therefore we use parallel gateways. There is only one start and end event in this process and we diagram them accordingly.
- We associate events, tasks and gateways with sequence and message flows.
- The last step for business modeling is defining inputs and outputs. In this phase of PM4E, we define the inputs and outputs using BPMN2.0 data input, output and store elements.

At the end of this phase, we have the following business model, depicted in Figure 4.8, in BPMN2.0 notation. Eclipse BPMN Modeler automatically checks the syntactic behaviour of the model, therefore there are some errors and warnings in the diagram. However, according to PM4E, this step is mainly for business users and we accept syntactical errors as long as the semantics is correct.

4.3.4.3 Execution Modeling

In this phase, we switch to execution modeling perspective to define the execution parameters and data modeling.

- Preparation: In this case study, we have selected Activity BPM Platform to execute the processes. Therefore the requirements for execution are fulfilled according to this suite's appropriate languages and notations.
- User Interfaces: In this process users complete their tasks and escalate the process to the next step. Therefore Activiti's forms extension for assigning forms to tasks is used. To be able to define the user interfaces, we describe the required execution parameters and variables, as can be seen in Figure 4.9

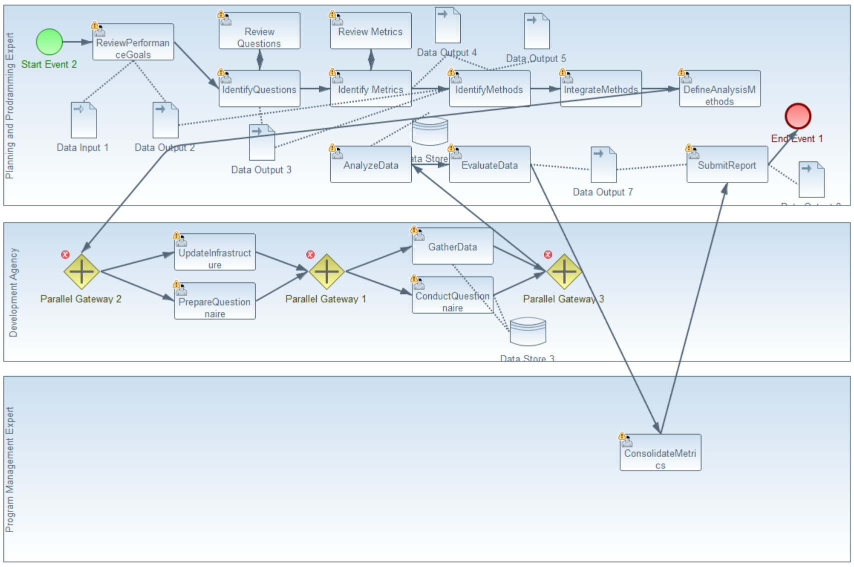
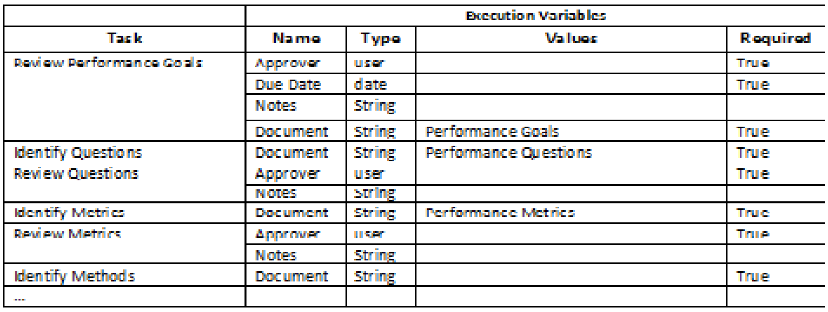


Figure 4.8: Business Model with BPMN2.0 Notation



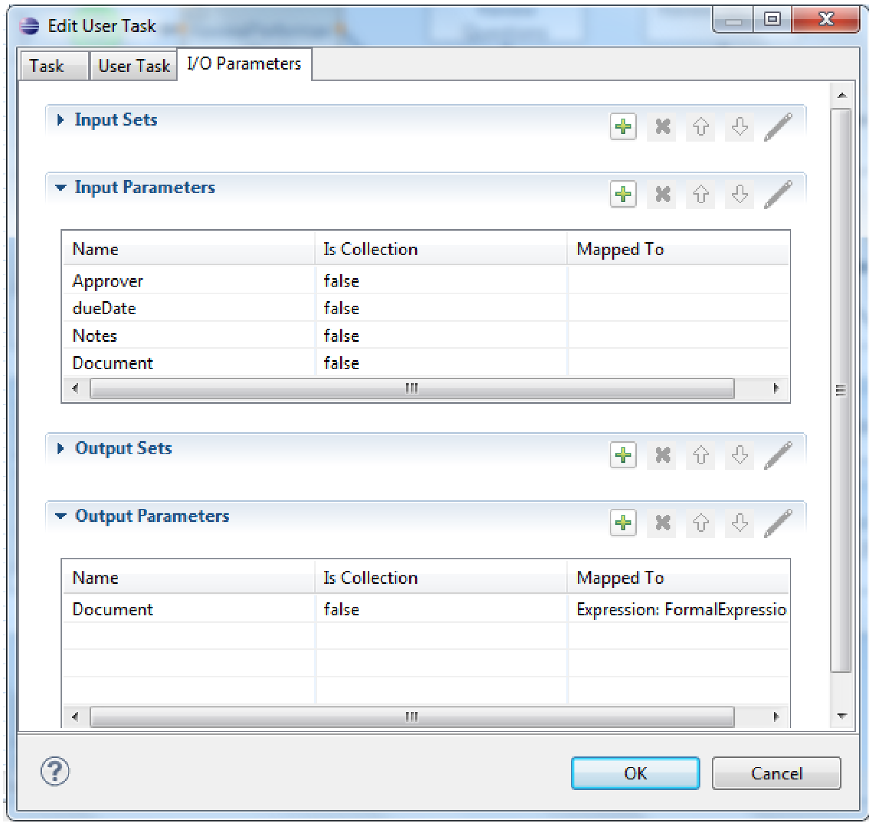
	Execution Variables			
Task	Name	Type	Values	Required
Review Performance Goals	Approver	user		True
	Due Date	date		True
	Notes	String		
	Document	String	Performance Goals	True
Identify Questions Review Questions	Document Approver	String user	Performance Questions	True True
	Notes	String		
Identify Metrics	Document	String	Performance Metrics	True
Review Metrics	Approver	user		True
	Notes	String		
Identify Methods	Document	String		True
...				

Figure 4.9: Case 2 User Interface Definition

- Data Definitions: Data definitions are handled while preparing user interfaces. In this step, we convert the data objects to required data definitions using data types provided by Activiti, as can be seen in Figure 4.9. These definitions are integrated into the diagram using Eclipse BPMN Modeler's task property window as can be seen in Figure 4.9.

- Model Overview: Below are some of the changes done in the business model to fulfill the execution requirements.

  - Add a timer definition to start event since it is triggered with a calendar.
  - Parallel gateways are updated by providing default paths. Activity requires extensionElements of BPMN2.0 to implement form design. For each task, if there are any inputs or outputs required by the user, we define following extension points



Name	Is Collection	Mapped To
Approver	false	
dueDate	false	
Notes	false	
Document	false	

Name	Is Collection	Mapped To
Document	false	Expression: FormalExpressio

Figure 4.10: Case 2 Data Definitions

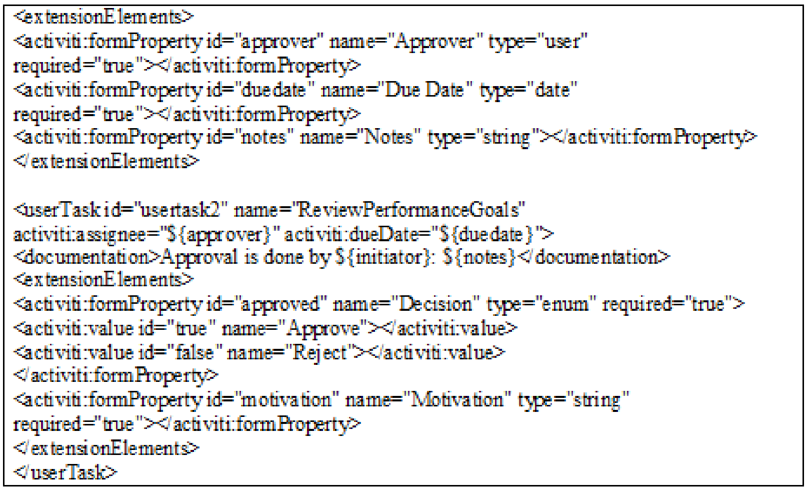
in the model so that Activity engine can interpret the task with its user interface. In this step, analogous to software user interface design, we design the user interfaces with process variables, as can be seen in Figure 4.9. These definitions must be done for each acitivity, task and gateway.

- Role Integration: There is no role integration in Activiti, users must be defined in the Activiti BPM platform.
- Execution: In this final step we have an executable process mode as depicted in Figure 4.12, which should be deployed to a server.

To deploy process to Activiti, the business model should be converted to a .bpmn20.xml format, which is listed in Appendix A. Then this model is imported into Activiti using manager console.

4.4 Case Study 3: Human Resources Planning

This case study was conducted for two purposes. One of them is to verify that PM4E methodology maintains the up-to-date status of the processes when a change occurs in the process



```
<extensionElements>
<activiti:formProperty id="approver" name="Approver" type="user"
required="true"></activiti:formProperty>
<activiti:formProperty id="duedate" name="Due Date" type="date"
required="true"></activiti:formProperty>
<activiti:formProperty id="notes" name="Notes" type="string"></activiti:formProperty>
</extensionElements>

<userTask id="usertask2" name="ReviewPerformanceGoals"
activiti:assignee="${approver}" activiti:dueDate="${duedate}">
<documentation>Approval is done by ${initiator}: ${notes}</documentation>
<extensionElements>
<activiti:formProperty id="approved" name="Decision" type="enum" required="true">
<activiti:value id="true" name="Approve"></activiti:value>
<activiti:value id="false" name="Reject"></activiti:value>
</activiti:formProperty>
<activiti:formProperty id="motivation" name="Motivation" type="string"
required="true"></activiti:formProperty>
</extensionElements>
</userTask>
```

Figure 4.11: Case 2 User Interface Transformation

definition. The second focus is on analyzing the methodology's efficiency with different modeling notations.

4.4.1 Background

This case study also uses a process definition from the same organization of second case study. The main objective of this process is conducting the annual human resources planning of the organization. The process also includes updating human resources strategy, job descriptions, personnel qualifications and planning for auxiliary personnel requirements.

4.4.2 Case Study Plan

The planned list of activities for this case study is listed below.

1. Develop business model with PM4E methodology
   - Input: Description of as-is processes
   - Data to be collected: Metrics on process details such as number of tasks, gateways and events
   - Data collection method: Metric collection
2. Develop executable model with PM4E methodology
   - Input: Business process model
   - Data to be collected: Analysis of neglected information from business models, number of additional process elements

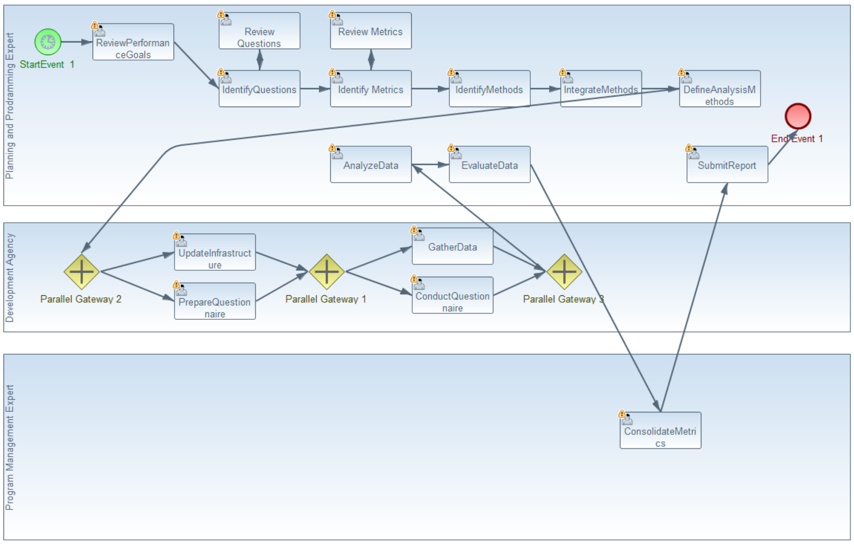


Figure 4.12: Executable Process Model

- Data collection method: Direct observation, metric collection

3. Produce executable outputs for the specified process engine

    - Input: Executable process model
    - Data to be collected: Time required for transformations
    - Data collection method: Metric collection

4. Implement a change in the executable model with PM4E methodology

    - Input: Executable process model
    - Data to be collected: Time required to implement the change
    - Data collection method: Metric collection

5. Produce executable outputs for the implemented change for the specified process engine

    - Input: Executable process model
    - Data to be collected: Time required for transformations
    - Data collection method: Metric collection

6. Analyze business model in terms of the implemented change

    - Input: Business process model
    - Data to be collected: Effect of the change on business model
    - Data collection method: Direct observation

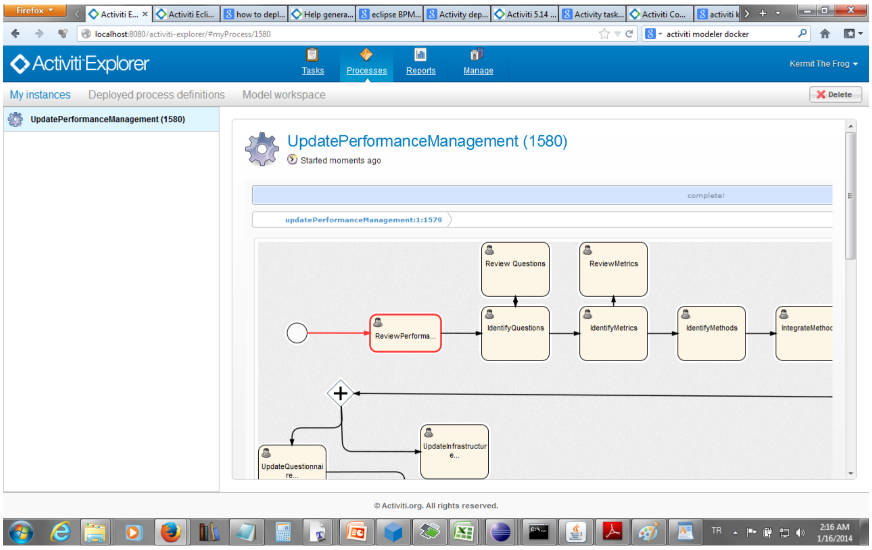


Figure 4.13: Deployed Process Model

4.4.3 Sources of Evidences

In this case study we have used the following sources of evidences, to analyze and evaluate the results.

- Documentation: In this case study, we use existing process definitions expresses in EPC notation to understand the as-is process.
- Physical Artifacts: We have executed the process models and therefore produced the following outputs:
  - Business process models, with EPC and BPMN notation
  - Interface and Data Definitions
  - Executable process models in BPMN20 format
  - Effort Data in terms of man-hours
- Direct Observation: We use direct observation to evaluate the process elements which are discarded in the executable process level.

4.4.4 Case Study Conduct

In this case study, we have used original eEPC models and transferred them into Eclipse BPMN2.0 Modeler to continue with execution modeling. Execution is realized via Activiti Business Process Management Platform. Our objective was to prove that PM4E may be used independent of the business modeling notation. Furthermore, we have generated a scenario where a change occurs in the process definition. We have updated the execution model and validate that the business model is also updated. The objective was to show that the process models with PM4E methodology maintains the up-to-date status of the process model for both business and IT users.

4.4.4.1 Business Modeling

The PM4E business modeling phase is conducted as follows:

- Identify process goals: This process aims to conduct human resource planning for the upcoming year.
- Identify roles: There are three participants defined for this process:
  - General secretary is the major actor in this process preparing the report, updating the strategy and job descriptions and planning for additional personnel.
  - Board of Directors review the human resource planning strategy.
  - Human resources update the personnel qualifications.
- Major activities in the process are defined in Table 4.7, in the actual sequence of the process. We also identify the start and end events in the process.
  - Start: Process is started when the time for annual human resources management has come.
  - End: Process ends with the identification of auxiliary personnel need for the upcoming year.

    In the following Table 4.7,

Table 4.7: Case 3 Activity Definitions

Activity	Role
Prepare annual report	GS
Review report	BD
Update human resources strategy	GS
Conduct work analysis	GS
Update job descriptions	GS
Review human resources planning	GS
Update personnel qualifications	HR
Identify personnel need	GS
Identify personnel role changes	GS
Identify auxiliary personnel	GS

- Identify major inputs and outputs: At this point we have the major activities with the roles who perform them. In this step we identify the input output relations, as depicted in 4.8.

Table 4.8: Case 3 Activity Definitions with Inputs and Outputs

Activity	Input	Output
Prepare annual report	Effort and Personnel Evaluation Reports and Personnel Satisfaction Report	Annual Report
Review report	Annual Report and Agency Activity Report and Regional Plans	
Update human resources strategy		HR Strategy
Conduct work analysis	Work Definitions and Annual Report	
Update job descriptions		Process Definitions
Review human resources planning	Process Definitions and Annual Report	
Update personnel qualifications		Personnel Qualifications
Identify personnel need	Need Definitions	
Identify personnel role changes		Agency Organizational Schema
Identify auxiliary personnel		Auxiliary Personnel Definitions

- Define conditions: No conditions apply for this process.
- Review Process: The process is reviewed by all participants in order to ensure that all roles in the process analysis step exists in the process model, all activities are connected with roles and all input and outputs are associated with the activities.

Once all the required information is gathered, we have modeled the process diagram with eEPC using ARIS Business Architect. ARIS Business Architect provides a facility to generate a BPMN model. The resulted model is helpful, however it is very poor. In this case, we needed to update the produced model using Eclipse BPMN2.0 Modeler.

4.4.4.2 Execution Modeling

After defining the user interface and data definitions, we have finalized execution modeling as depicted in Figure 4.14.

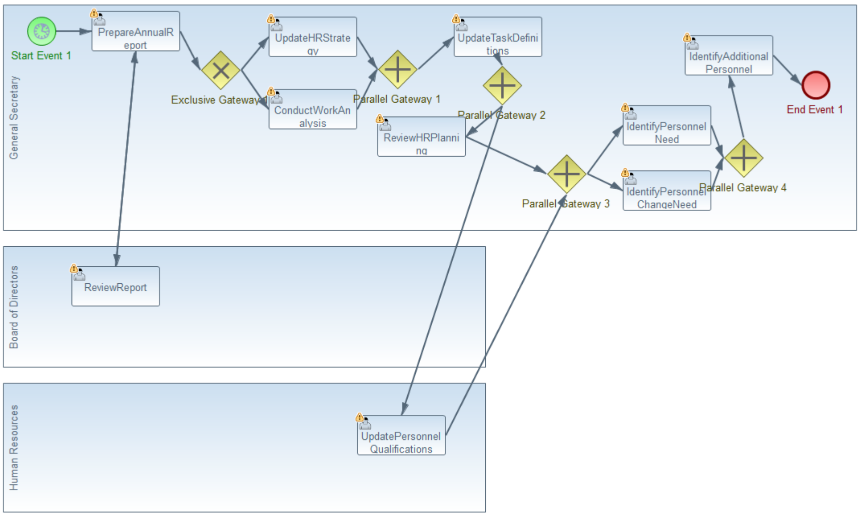


Figure 4.14: Case 3 User Interface and Data Definitions

4.4.4.3 Change in the Process

In this scenario, we add an additional step for approving the changes made in personnel qualifications. When we open the process in business user perspective we see that the change in the process is reflected to business perspective, as depicted in Figure 4.15. BPM supports managing changing in business process by providing a flexible and more adaptable IT infrastructure. PM4E, utilizing this opportunity provided by BPM, provides a method to represent business processes which may be more adaptive to changes in business processes. Since the change is done once, it is proactive and the changes are reflected to both perspectives in real time. The process model is kept ”alive” as the real-life business process.

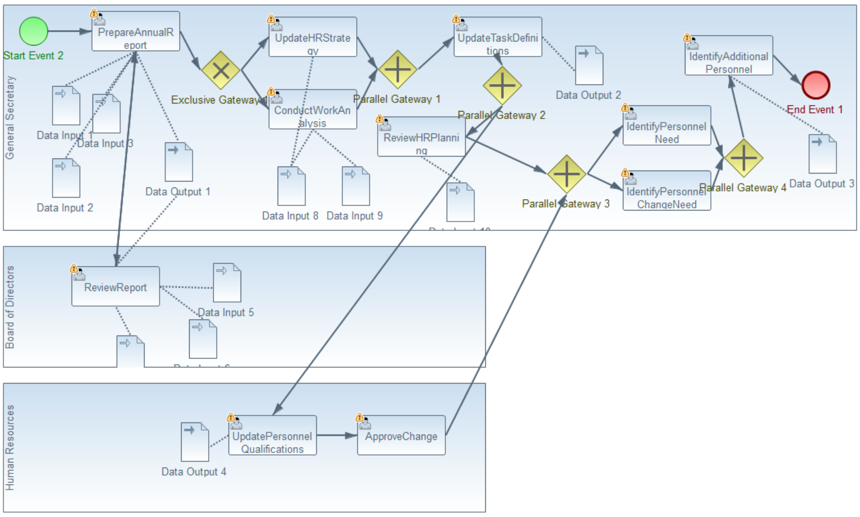


Figure 4.15: Case 3 Updated Business Process Model

4.5 Discussions

In the previous sections, we described how PM4E can be applied in different case studies. In the first case study, we have identified the requirements of execution modeling and the components required for execution. We have identified the challenges that are presented for a business model to be executable. This study also revealed the fact that before starting modeling, there are decisions to make in order to establish a common understanding of the process. We have added a premodeling phase to make sure that each stakeholder’s requirements will be met in the final executed process definition. We have applied our findings in the second case. This case study lead to exploring the usability of different notations which resulted in the conduct of third case study.

In next subsections, we summarize our findings regarding to our claims.

- First case study revealed the requirements for an executable process model. Most of the BPM suites offer either a stand-alone application or an Eclipse plug-in for process modeling. The Eclipse environment does not offer a user-friendly graphics interface and it is not easy to visualize the model, especially for end users who are not familiar with programming interfaces. Although stand-alone applications for business modeling are more user friendly, overall, the modeling environments which BPM suites offer are limited when compared to tools which are specifically designed for business process modeling, such as ARIS [ref1-9]. For example, the hierarchical organization of the processes is not permitted in BPM suites, in this case modelers have to deal with the whole process at once, which is not easy to handle. Moreover, BPM suites support BPMN notation, however there are some problematic areas. Since the execution perspective is dominant throughout the modeling phase, there are some controls that make it very difficult to express the circular flows in the processes. Another issue is role definitions expressed as lanes in BPM suites. Defining role interactions may be quite difficult since each interaction requires the definition of execution parameters. In the business model, the focus is on understanding, therefore elements which are not immediately relevant to processes but enhance understanding are permitted. However, in execution models the focus is on execution, therefore, these kinds of elements may be omitted and some knowledge may be lost, not all aspects of the business model can be reflected in the execution model. Therefore, the selection of modeling environment is essential for process execution.

Executability check for execution modeling is very helpful for IT modelers. Intalio provides executability check during modeling, however, the support for resolving issues is very difficult and modelers need a high level of technical knowledge. For example, Intalio gives error messages for wrongly associated elements but it is necessary to have BPEL knowledge to understand what the error refers to. In other suites, modeling phase is separated from execution phase; therefore execution check is not immediately available.

The final step in the study was to execute the business models and it proved to be the most time consuming and difficult stage. All BPM suites use web technologies to execute the processes via Microsoft, Java or other proprietary solutions. Setting up the environment is time consuming and very difficult, since each suite has its specific requirements, for example, Intalio requires Java, Eclipse and J2EE knowledge to run the server. BizAGI requires familiarity with Microsoft technologies and Web-methods require high RAM and extensive knowledge of the suggested framework. Suites have different approaches for defining execution parameters. Some suites have separate definitions for each phase. Users model the process first, and then, later, add roles, inputs, outputs and relations with other organizational services. This approach is easier since it separates the modeling phase from execution, reducing the complexity of modeling. However, some tools require definition of forms, inputs, outputs while modeling, which re-quires both programming and BPEL knowledge. For example, there is more work to define conditions for expressing loops and exclusive gateways, and this cannot be done without technical knowledge and experience.

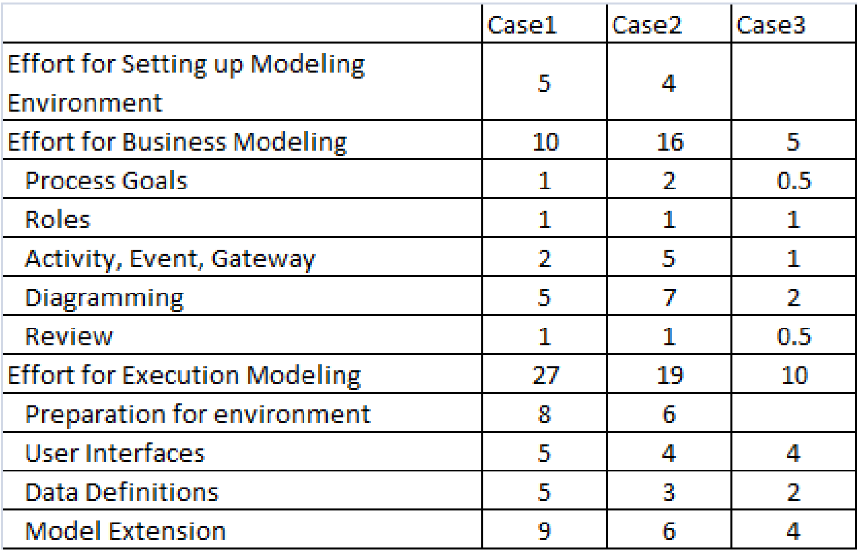
- Maintaining a common understanding of process descriptions between business and IT users: Process modeling practices can be implemented by a vast amount of users for a vast amount of purposes. In the context of this study process models are intended for process description for business users and process execution via an execution engine for IT users. When different characteristics of these groups are considered, it is natural that different requirements will be needed in the modeling phase. BPMN 2.0 is extensively used for this purpose, not because of it is superior to other notations, but mainly because it is supported by many vendors and open-source software. BPMN 2.0 provides many elements to define the processes in graphical notation, as well as defining execution variables. However, although it provides additional capabilities for IT users, having too many types of elements makes it confusing especially for business users. In its current state, BPMN cannot bridge the gap between business and IT alone since modeling is more than notation. PM4E offers that when modeling is done with the right abstraction level, the process models can be used for descriptive purposes for business people and as execution models for IT users. PM4E includes the concerns for both user groups and proposes a methodology for building a common view of the processes usable for both user groups.

Using descriptive process models for execution has two main goals, which actually contradict each other. The models should be simple enough to be understandable by different participant to reach an agreement. They also should meet requirements of formal data execution modeling semantics, which is a cause for increased complexity. This complexity decreases the possibility of achieving an agreement. In other words, even inexperienced users should be able to understand the process with the process model, but at the same time it should contain necessary details required for execution. The business and execution perspectives offered in the business modeling phase bridges this gap between business and IT domains by overcoming this contradiction. Moreover, PM4E methodology guides all stakeholders to agree on the very basic attributes of the process models such as its goals, major users, major inputs and outputs. PM4E business modeling phase allow business users to focus on descriptive aspects of the process model and execution modeling phase guides IT users to focus on executions parameters, without loosing the descriptive properties which are the most basic aspects maintaining common understanding.

- Reduction in total effort: The main contribution of PM4E, in terms of effort, is eliminating the need to re-model the business processes for description and execution. Table 4.9 summarizes the total effort required to model the processes, in person days, including setting up BPM modeling environments and the learning phase as well as the effort required to execute the processes including setting up BPM execution environments and learning phase.

  When we compare the modeling effort, in person days, in terms of only business modeling phase, we see that PM4E requires more time, as depicted in Figure 4.16. It is seen that modeling with PM4E requires more time than modeling with eEPC, which was the selected notation for modeling the processes in the selected organization. There are several reasons for this surplus. Proficiency in the selected environment is one of the reasons. The modelers using eEPC were fluent and competent in modeling with this notation. Another reason is the inadequacy of process modeling environment selected for the case studies. As claimed in a previous study, process modeling environments designed to support process execution generally address to IT people and there are not user-friendly to model the processes [14].Another important reason is the fact that during the modeling phase, many aspects between IT and business people are decided on which brings overload to modeling the process.

Table 4.9: Effort for modeling and execution in person days



	Case1	Case2	Case3
Effort for Setting up Modeling Environment	5	4	
Effort for Business Modeling	10	16	5
Process Goals	1	2	0.5
Roles	1	1	1
Activity, Event, Gateway	2	5	1
Diagramming	5	7	2
Review	1	1	0.5
Effort for Execution Modeling	27	19	10
Preparation for environment	8	6	
User Interfaces	5	4	4
Data Definitions	5	3	2
Model Extension	9	6	4

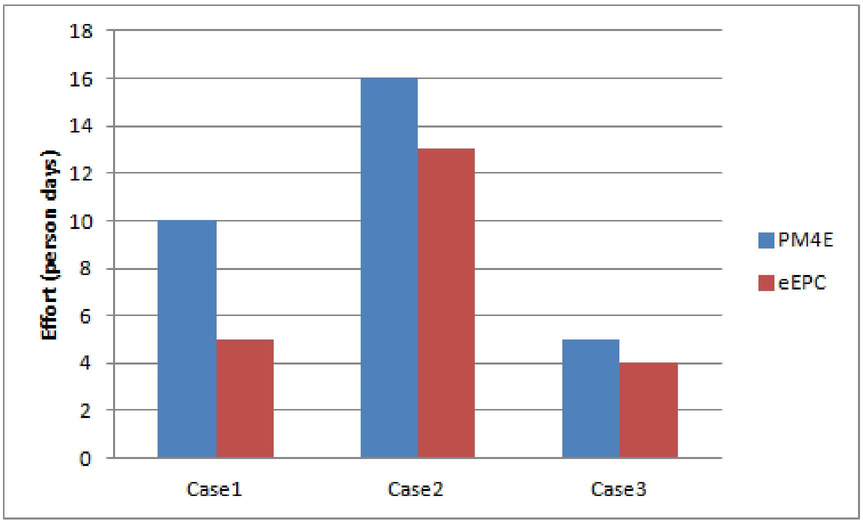


Figure 4.16: Effort for business modeling

In order to compare the total effort, we have made some estimations based on the functional size measures of the processes, since there was no data for actual implementation efforts. The estimations are done assuming that the processes will be implemented from scratch using ”waterfall” software development method. Basing on these assumptions, total effort is calculated as the sum of efforts required for requirement analysis, design, implementation and test phases. System requirements analysis were calculated to be 211 and 195 COSMIC Function Points. We have transformed these size measurements into effort data. This transformation is done utilizing Capers Jones data, which have been accepted as a de-facto standard for software size estimation in the industry [90].

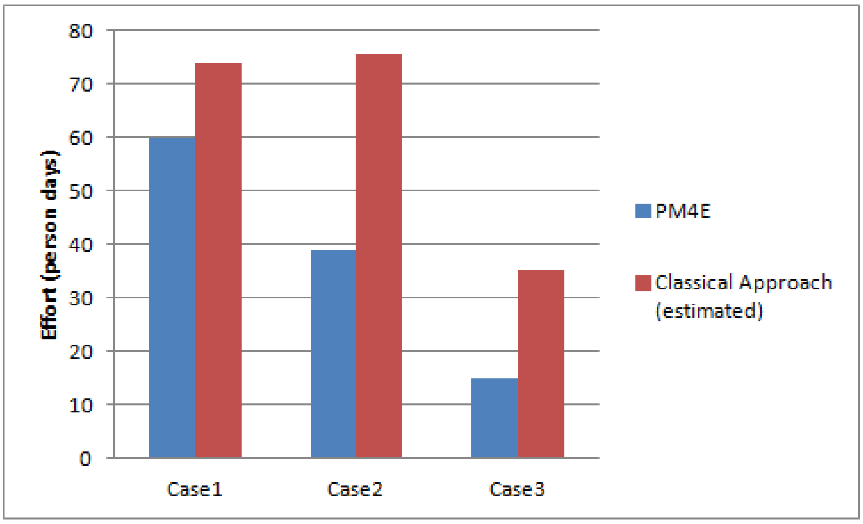


Figure 4.17: Effort for implementation

We have selected benchmark data from Capers Jones for projects which are similar to the processes modeled and using these benchmark data we have estimated the effort required to fulfill the requirements of the process. Although, the actual efforts would change according to the team size of development team, we believe they would satisfy our intention to show the comparison between classical approaches and PM4E. In Figure 4.17, the comparison of total efforts - from modeling to deployment - are presented. As the figure presents, despite business modeling phase requires more time, reusing the process model provides great improvements in terms of total effort. Our studies show that the effort is estimated to be reduced by 18.36 percent in the average. This finding is consistent with the estimations of Gartner Group, which claims that BPM initiations can save up to 20 percent for cost savings [34].

We have also compared the productivity required to execute processes with and without PM4E, considering the sizes of the processes, as can be seen in Figure 4.18. We have assumed that function point measurements can be used to size processes, therefore modeling productivity can be mesaured as FP / person day. In the first case study, PM4E was not yet developed, therefore we have not used the methodology. Although the sizes of the processes modeled in each case study was different than the other studies, the figure reveals that the productivity increases with the use of our methodology. In the first case study, the productivity is low compared to second and third syudies. In second and third case studies, we have modeled the processes from business modeling to execution modeling phase using PM4E methodology and the results show a significant increase in the productivity.

- Providing complete process definitions: PM4E considers the requirements of all involving parties. When following the modeling phases, the inconsistencies with the originally defined models are revealed. The questions asked by PM4E reveals many

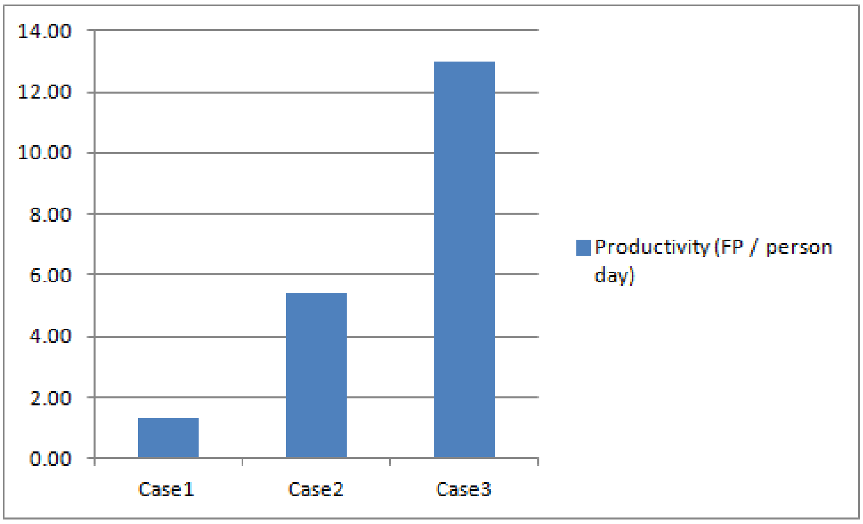


Figure 4.18: Comparison of Effort with and without PM4E

details which are generally not expressed by business users. Moreover, the checklist provided by PM4E considering roles, activities and outputs enhances solving many of the ambiguity and incompleteness issues in the process.

- Maintaining up-to-date process definitions: Many process models suffer the fact they become obsolete once they serve their purpose. The case is very similar to software development; once necessary models or document are produced in order to develop software, most of the time they are discarded. It is very costly, especially in terms of schedule, to update these artifacts and therefore they become discarded. Same rules apply in business process modeling and management. As Branco et.al states, currently, major process modeling tools do not provide means to support consistency among produced models [12]. PM4E overcomes this challenge by focusing on process model as a major output of the methodology. Since business and executable models are the same model but with different representations, a change performed in either of them assures that the model description is up-to-date.

- Tool dependency: PM4E defines a methodology for escalating from descriptive process models to executable process models. The case studies reveal that the methodology is applicable regardless of the selected notation since the major contribution is guiding users to define what is to be modeled and how. However, process execution relies mainly on the specifications provided by the process engines which use many different notations to orchestrate various components. Accomplishing the execution via a process engine requires deep knowledge of the technologies and the methodologies used. Currently, notation conversion capabilities are far from being perfect, therefore we had to manually edit the process definitions.

- Producing executable models from descriptive process models requires integration of various components, most of which require transformation between various notations.

Although there are many studies and products focusing on this subject, current tools are not mature enough to provide full compatibility. This incompatibility resulted in the concept of using adapter models. For example, processes modeled in eEPC cannot be reused in any of the BPM suites. There are implications that BPMN specifications can be imported into some suites but a one-to-one transformation cannot be achieved. In order to execute processes with BPM suites, modelers have to re-model the processes in BPMN notation using the modeling environment that the suites offer.

4.6 Threats to Validity

Due to the wide span of the concepts spanned in process execution, some possible threats to validity may occur. We list the major areas of threats and the precautions to avoid them as follows:

- Construct Validity makes the case study conductors think carefully and thoroughly on the measures so that the case study process is appropriate to evaluate the method [91]. In order to overcome this threat, the case study plan is reviewed after the exploratory study and the metrics used for reaching conclusions are revised. We have only used effort data to compare te outputs of PM4E with classical approaches, since we did not have any other additional measures. The improvements should also be defined and measured in terms of cost reduction provided by executing the processes using descriptive process models.

- Internal Validity is defined as the degree to which changes in one variable causes the changes in another variable and there is sufficient evidence for the relation [91]. Since we derive conclusions based on the results of the methodology, internal validity becomes a concern to validate the results of the methodology. The validity of the conclusions depend on the quality of application of PM4E. In order to avoid this threat, the case study design provides a chain of evidences for each case study. These evidences include which data to collect and when to collect them. Another measure taken to avoid internal validity threats is the randomization of the test cases. The selected case studies conform to selection criteria however each one of them differ in their properties reducing the possibility to be biased in case studies which decreases the possibility of an internal validity threat.

  Another threat for internal validity is the lack of involving other researchers to use PM4E methodology to model their processes for execution. In our case studies, we - being the developers of PM4E methodology - have modeled the processes both in business and execution modeling phases. In this respect, we can say that we have used action research in order to show that our methodology improves the effort required for business process execution. An important threat is the possibility of research conductor's being biased. Also, the familiarity with selected business process modeling notations may affect the results since the results are compared in terms of effort required for modeling in selected business process modeling notations.

- External Validity considers the extent to which the results of the study is generalizable [91]. In order to overcome this threat, we have engaged in a multiple case study, where the method is replicated several times and the results are validated. However, in our study, we did not have access to an organization which actually modeled their processes for execution. In order to fully show that PM4E methodology can be used for various processes in various organizations, this study should be replicated in an organization having a process execution infrastructure. The processes should be modeled and executed with PM4E methodology and the results should be compared to the way the organization executes their processes. In this way, it would be possible to evaluate how useful is PM4E methodology for improving organization's processes.

  Process selection criteria is another threat to our study. We have identified process selection criteria for the case studies and applied the methodology to model the processes fitting these criteria. In order to demonstrate the generalizability of our methodology, the processes which do not fit our criteria should be modeled and executed, so that the variation can be observed.

CHAPTER 5

CONCLUSIONS

Organizations engage in process modeling activities for many purposes. In this study we focus on process modeling purposes for defining the ”as-is” processes and executing them. As the purposes differentiate, the notations and method for process modeling diverge in order to fulfill the requirements of the purpose. Consequently, this results in development of more and more types of notations. When organizations use these differentiated notations, the general case is that various models for different purposes are constructed within the organization for the same process. This situation poses consistency issues within the organization, since a change in the process should be reflected to all of the models. Moreover, it requires significant amount of time and effort to produce the process models for different purposes.

A unified approach which combines the requirements for defining the processes with a high-level point of view and for producing detailed executable models enhances the reusability of the model, therefore decreases the total effort for process modeling. Moreover, such an approach would increase the understanding among various stakeholders of the process within the organization. This would result in better alignment of business and IT which increases the organization’s competitiveness.

The literature review shows that there is no methodology to support such an approach. Studies focusing on process modeling for descriptive purposes neglect the requirements of process execution. On the other hand, process execution modeling methods are too complex and comprehensive for inexperienced users to understand. Moreover, most of the studies focus on the notations and their usage. However, utilizing the notations in a methodical manner is more important than the notations used. This study focuses on the challenges arising from the unification of descriptive and execution process models. It provides a method that enables organizations to model their processes both with business and IT perspectives.

In this chapter, we summarize the conclusions and suggest future research areas as well as the limitations of the study.

5.1 Contributions

During the conduct of this study, we have inspected many BPMSs and identified difficulties and challenges proposed by them. These studies resulted in several papers contributing the literature.

The major contribution of this study is the development of PM4E process modeling methodology which is designed for utilizing descriptive process models for execution purposes. The objective of this research is to develop a methodology for organizations, which will involve business process improvement and business process management initiations, to enable process modeling in a way that will be used by both domains. The method facilitates business and IT alignment by creating a common understanding as well as increasing the consistency of the process models.

In order to validate the applicability of PM4E, a multiple-case study with three cases were planned and conducted. The first case was mostly an exploratory case study to identify the requirements of a unified process modeling approach. In this study we have inspected several BPMS and identifies major challenged to execute process models. The second and third case studies were conducted to examine the applicability of the method. The findings of the case studies form the basis for this study's contribution in the field of process modeling.

The case studies show that when modeling is done with the right abstraction level, the process models can be used for descriptive purposes for business people and as execution models for IT users. PM4E includes the concerns for both business and IT user groups and proposes a methodology for building a common view of the processes usable for both user groups. Using descriptive process models for execution has two main goals, which actually contradict each other. The models should be simple enough to be understandable by different participant to reach an agreement. They also should meet requirements of formal data execution modeling semantics, which is a cause for increased complexity. The business and execution perspectives offered in the business modeling phase bridges this gap between business and IT domains by overcoming this contradiction. Moreover, PM4E methodology guides all stakeholders to agree on the very basic attributes of the process models such as its goals, major users, major inputs and outputs. This guidance not only enhances a common understanding, but also aids the users in the subsequent phases of process execution.

Another contribution of this study is decreasing the total time required for modeling organization's processes. In its current state, many organizations keep different models for different purposes. This practice requires significant time for each model to be produced. PM4E guides users to share the effort among various stakeholders which reduces the re-work of the degree of process modeling resulting in lower effort. Moreover, since there is only one model for the same process, the effort required to model the changes in the process is also decreased.

PM4E considers the requirements of all involved parties which increases the completeness of the process models. When following the modeling phases, the inconsistencies with the originally defined models are revealed.

Many process models suffer the fact they become obsolete once they serve their purpose. PM4E overcomes this challenge by focusing on process model as a major output of the methodology. Since business and executable models are the same model but with different representations, a change performed in either of them assures that the model description is up-to-date.

This study shows that descriptive process models can be executed, however, it also presents the fact that assuming the models which are produced by business users can be directly executable is far from reality. The major cause of this false assumption results from the wide area that is spanned by process execution. Executable processes are, by nature, need to be very complex to be directly understandable by business users. However, as our study reveals, this complexity may be overcomed by defining different perspectives with different abstraction levels and presenting the model to the users within these perspectives.

Overall, we can conclude that this study reveals the evidence that deriving executable process models from descriptive process models is a valuable approach for organizations.

5.2 Limitations and Future Work

A major limitation of our study is using the processes of a single organization which actually was not involved in process execution initiations. Therefore, we did not have the opportunity to compare the results to actual process execution efforts. We have made estimations basing on the properties of the processes using commonly utilized estimation techniques. Lacking of effort data to compare results is a limitation in this study, which may be overcomed by applying PM4E methodology in an organization utilizing process automation initiatives.

Producing executable models from descriptive process models requires integration of various components, most of which require transformation between various notations. Although there are many studies and products focusing on this subject, current tools are not mature enough to provide full compatibility. In this study, we had to manually edit some of the produced xml outputs in order to make them compatible in the environment they are to be transported. Studies regarding notation conversions are still required in this area, despite the fact that there are many attempts on this subject.

No BPM vendor provides a general purpose process engine that business users can use to create a solution from descriptive process models. Accomplishing the execution via a process engine requires deep knowledge of the technologies and the methodologies used. Currently, notation conversion capabilities are far from being perfect, therefore we had to manually edit the process definitions.

We have left some aspects of process execution out of the scope of this thesis such as integration with business rules. We propose that extending PM4E to include means for integration with business rule management systems would be beneficial.

REFERENCES

[1] Gartner group, meeting the challenge: The 2009 cio agenda. Technical report, Gartner Executive Program Report, 2009.

[2] M. A. Ould. *Business Processes: Modelling and Analysis for Reengineering and Improvement*. Wiley, New York, 1995.

[3] J. Freund, B. Rucker, and T. Henninger. *Praxishandbuch BPMN: Incl. BPMN 2.0*. 2010.

[4] J. Mendling, H. A. Reijers, and W. M. P. van der Aalst. Seven process modeling guidelines (7pmg). *Information Software Technology*, 52:127–136, 2010.

[5] H. A. Reijers, J. Mendling, and J. Recker. Handbook on business process management. Technical report, Springer Verlag.

[6] R. Hung. Business process management as competitive advantage: a review and empirical study. *Total Quality Management Business Excellence*, 17:21–40, 2006.

[7] S. Lippe, U. Greiner, and A. Barros. A survey on state of the art to facilitate modelling of cross-organisational business processes. In *Proceedings of the 2nd GI-Workshop XML4BPM*, 2005.

[8] G. Phifer, S. Hayward, and D. Flint. Technologies that enable business process fusion. Technical report, Gartner, 2003.

[9] J. Recker, M. Rosemann, M. Indulska, and P. Green. Business process modeling - a comparative analysis. *Journal of the Association for Information Systems*, 10(4), 2009.

[10] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske. Business process management: A survey. In *Proc. Business Process Management: International Conference*, 2003.

[11] M. Melenovsky and J. Sinur. Having a bpm maturity model is important for long-lasing bpm success. *Gartner, Business Integration Journal*, 2006.

[12] M. C. Branco, K. Czarnecki, Y. Xiong, J. Kuster, and H. Volzer. A case study on consistency management of business and it process models in banking. *Software and Systems Modeling*, 2013.

[13] P. G. Armour. *The Laws of Software Process: A New Model for the Production and Management of Software*. Auerbach Publications, 2004.

[14] F. C. Yesildoruk and O. Demirors. *Utilizing Process Definitions for Process Automation: A Comparative Study*. BPM and Workflow Handbook. 2010.

[15] T. H. Davenport. Process innovation: Reengineering work through information technology. 1993.

[16] M. Hammer and J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business., NY, 1993.

[17] K. Terai, N. Izumi, and T. Yamaguchi. Coordinating web services based on business models. In *Proceedings of the 5th international conference on Electronic commerce*, volume 50, pages 473–478, 2003.

[18] A. Lindsay, D. Downs, and K. Lunn. Business processes - attempts to find a definition. *Information and Software Technology*, 45:1015–1019, 2003.

[19] K. C. Laudon and J. P. Laudon. *Management Information System: New Approaches to Organization Technology*. Prentice Hall International, Inc, 1998.

[20] *Improving Business Processes*. Harvard Business School Press, 2013.

[21] S. Sorumgard. Verification of process conformance in empirical studies of software development., 1997.

[22] M. Indulska, P. Green, J. Recker, and M. Rosemann. Business process modeling: Perceived benefits. *Lecture Notes in Computer Science*, 5829:458–471, 2009.

[23] A. W. Scheer and M. Nuttgens. Aris architecture and reference models for business process management. *Business Process Management, Models, Techniques, and Empirical Studies*, pages 376–389, 2000.

[24] A. Lodhi, V. KÃ¶ppen, and G. Saake. Business process modeling: Active research areas and challenges. Technical Report ISSN 1869-5078, Elektronische Zeitschriftenreihe der Fakultat fur Informatik, 2011.

[25] R. Lu and S. Sadiq. A survey of comparative business process modeling approaches. *Business Information Systems, LNCS*, 4439:82–94, 2007.

[26] A. Fleischmann, S. Rass, and R. Singer. *S-BPM Illustrated: A Storybook about Business Process Modeling and Execution*. Springer, 2013.

[27] M. Cumberlidge. *Business Process Management with JBoss jBPM: A Practical Guide for Business Analysts*. Packt Publishing, 2007.

[28] J. Clemmer. Process reengineering and process improvement: not an either/or choice. *CMA Magazine*, 68:36–39, 1994.

[29] U. S. Bititci and D. Muir. Business process definition: a bottom-up approach. *International Journal of Operations Production Management*, 17(4):365–374, 1997.

[30] The workflow management coalition specification. Technical Report WFMC-TC-1011, Workflow Management Coalition, 1996.

[31] N. Leeming. Business process management implementation. *The Journal of Enterprise Architecture*, 1:69–91, 2005.

[32] G. Scheithauer, G. Wirtz, and C. Toklu. Bridging the semantic gap between process documentation and process execution. In *The 2008 International conference on Software Engineering and Knowledge Engineering*. Meghan-Kiffer Press, 2008.

[33] J. Mendling. Empirical studies in process model verification. *T. Petri Nets and Other Models of Concurrency*, 2:208–224, 2009.

[34] Gartner exp worldwide survey. Technical report, Gartner, 2010.

[35] H. Smith and P. Fingar. *Business Process Management, the third wave*. Meghan-Kiffer Press, 1st edition, 2003.

[36] H. Smith. The emergence of business process management. *Infirmation and Software Technology*, 45(15):1065–1069, 2003.

[37] M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag New York Inc., 2007.

[38] J. Mendling. Metrics for process models:empirical foundations of verification, error prediction, and guidelines for correctness. *Lecture Notes in Business Information Processing*, 6, 2008.

[39] B. Curtis, M. I. Kellner, and J. Over. Process modelling. *Communications of the ACM*, 35(9), 1992.

[40] J. Mendling, M. Nuttgens, and G. Neumann. A comparison of xml interchange formats for business process modelling. In *Proceedings of EMISA 2004 â In-formation Systems in E-Business and E-Government*, 2004.

[41] Business process modeling notation (bpmn) specification 2.0. Technical report, Object Management Group:, 2006.

[42] B. Silver. *BPMN Method and Style*. 2nd edition, 2012.

[43] BPMN. Object management group business process model and notation.

[44] C. L. McGowan and S. A. Bohner. Model based process assessments. In *15th International Conference on Software Engineering*, pages 202–211, 1993.

[45] Business process execution language for web services version 1.1. Technical Report BPEL4WS 1.1, OASIS, 2006.

[46] B. List and B. Korherr. An evaluation of conceptual business process modelling languages. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, 2006.

[47] S. A. White. *An Example of Using BPMN to Model a BPEL Process*. The Workflow Handbook. Future Strategies Inc, Lighthouse Point, FL, USA, 2005.

[48] R. Shapiro, S. A. White, C. Bock, N. Palmer, M. zur Muehler, M. Brambilla, and D. Gagne. *BPMN 2.0 Handbook Second Edition*. Future Strategies Inc., Fl, USA, 2 edition, 2012.

[49] C. Menzel and R. J. Mayer. The idef family of languages. *Handbook on architectures of information systems*, pages 215–249, 2006.

[50] P. Loos and T. Allweyer. *Process Orientation and Object-Orientation - An Approach for Integrating UML and Event-Driven Process Chains (EPC)*. Institut fur Wirtschaftsinformatik, Saarbrucken, Germany., 1998.

[51] IDEF. Integrated definiton methods.

[52] L. Tsironis, K. Anastasiou, and V. Moustakis. A framework for bpml assessment and improvement: A case study using idef0 and eepc. *Business Process Management Journal*, 15(3):430–461, 2009.

[53] M. A. Ould. Designing a re-engineering proof process architecture. *Business Process Management Journal*, 3(3):232–247, 1997.

[54] K. Phalp and M. Shepperd. Quantitative analysis of static models of processes. *Journal of Systems and Software*, 52(2-3):105–112, 2000.

[55] G. Abeysinghe and K. Phalp. Combining process modelling methods. *Information and Software Technology*, 39(2):107–124, 1997.

[56] N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and P. Wohed. On the suitability of uml 2.0 activity diagrams for business process modelling. In *Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling*, volume 53, pages 95–104, 2006.

[57] Agile Modeling. Uml 2 activity diagramming guidelines.

[58] W3C. Web services description language (wsdl) 1.1. Technical Report http://www.w3.org/TR/2001/NOTE-wsdl-20010315, W3C, 2001.

[59] H. Mili, G. B. Jaoude, Ã. Lefebvre, G. Tremblay, and A. Petrenko. Business process modeling languages: Sorting through the alphabet soup. *ACM Computing Surveys (CSUR) Surveys*, 43(1), 2010.

[60] T. Rozman and R. V. Horvat. Case study of the suitability of the process modelling methodology in a team. In *Proc. of 28th International Conference on Information Technology Interfaces*, pages 103–108, 2006.

[61] C. Carnaghan. Business process modeling approaches in the context of process level audit risk assessment: An analysis and comparison. *International Journal of Accounting Information Systems*, 7(2):170–204, 2006.

[62] R. Yamamoto, K. Yamamoto, K. Ohashi, and J. Inomata. Development of a business process modeling methodology and a tool for sharing business processes. In *12th Asia-Pacific Software Engineering Conference (APSEC'05)*, pages 679–686, 2005.

[63] Y. G. Kim. Process modeling for bpr: Event-process chain approach. In *Proceedings of ICIS*, 1995.

[64] A. Grosskopf, J. Edelman, and M. Weske. Tangible business process modeling - methodology and experiment design. In *Business Process Management Workshops*, pages 489–500, 2009.

[65] N. Desai, A. K. Chopra, and M. P. Singh. A methodology for modeling and evolving cross-organizational business processes. *ACM Trans. Softw. Eng. Methodol.*, 19(42), 2009.

[66] O. Turetken and O. Demirors. Process modeling by process owners: A decentralized approach. *Software Process: Improvement and Practice*, 13(1):75–87, 2008.

[67] A. W. Scheer. *ARIS - Business Process Modeling*. Berlin, Germany, 3 edition, 2000.

[68] D. Lubke, T. Luecke, K. Schneider, and J. M. Gomez. Using event-driven process chains for model-driven development of business applications.

[69] J. Ziemann and J. Mendling. Epc-based modelling of bpel processes: a pragmatic transformation approach. In *Proceedings of the 7th International Conference Modern Information Technology in the Innovation Processes of the Industrial Enterprises*, 2005.

[70] R. Bendraou, A. Sadovykh, M. P. Gervais, and X. Blanc. Software process modeling and execution: The uml4spm to ws-bpel approach. In *Proceedings of IEEE EUROMICRO 2007*, pages 314–322, 2007.

[71] W. M. P. van der Aalst. A general divide and conquer approach for process mining. In *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, pages 1–10, 2013.

[72] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

[73] R. Conradi, C. Fernstrom, A. Fuggetta, and R. Snowdon. Towards a reference framework for process concepts. In *In 2nd European Workshop on Software Process Technology (EWSPT'92)*, pages 3–17, 1992.

[74] W. Tscheschner. Transformation from epc to bpmn. Technical report, Oryx Research, 2010.

[75] E. Soderstrom, B. Andersson, P. Johannesson, E. Perjons, and B. Wangler. *(2002). Towards a Framework for Comparing Process Modelling Languages*. CAISE 2002, A. Banks Pidduck et.al. (Ed.). LNCS 2348. Springer-Verlag, Berlin, Heidelberg, 2002.

[76] Activiti. Activiti components.

[77] Eclipse. Eclipse bpmn2 modeler.

[78] J. Hamel, S. Dufour, and D. Fortin. *Case Study Methods*. Sage Publications, Inc, Newbury Park, CA, 1993.

[79] M. D. Myers. Qualitative research in information systems. *MIS Quarterly*, 21(2):241–242, 1997.

[80] R. K. Yin. *The Case Study Method: An Annotated Bibliography*. COSMOS Corporation, Washington, DC, 1983-1984 edition, 1983.

[81] R. K. Yin. *Case Study Research: Design and Methods*. Sage Publications, Inc, 2 edition, 1994.

[82] P. Hodkinson and H. Hodkinson. The strengths and limitations of case study research. In *Proceedings of the Annual Learning and Skills Development Agency Conference*, 2001.

[83] P. Verschuren. Case study as a research strategy: Some ambiguities and opportunities. *International Journal of Social Research Methodology*, 6(2), 2003.

[84] I. Benbasat, D. K. Goldstein, and M. Mead. *MIS Quarterly*, 11(3):369–386, 1987.

[85] J. W. Cresswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, Inc.

[86] A. Finkelstein, J. Kramer, B. Nuseibeh, L. Finkelstein, and M. Goedicke. Viewpoints: A framework for integrating multiple perspectives in system development. *International Journal of Software Engineering and Knowledge Engineering*, 2(1):31–58, 1992.

[87] A. Finkelstein and I. Sommerville. The viewpoints faq. . *Software Engineering Journal*, 11:2–4, 1996.

[88] B. Nuseibeh, J. Kramer, and A. Finkelstein. Viewpoints: meaningful relationships are difficult! In *Proceedings of the 25th international conference on Software engineering*, 2003.

[89] M. Verlage. Multi-view modeling of software processes. In *Proceedings of EWSPT3*, pages 123–127, 1994.

[90] T. C. Jones. *Estimating Software Costs*. McGraw-Hill, Inc, 1998.

[91] H. Seltman. Online textbook for experimental design for behavioral and social sciences.

APPENDIX A

Case Study Process Models

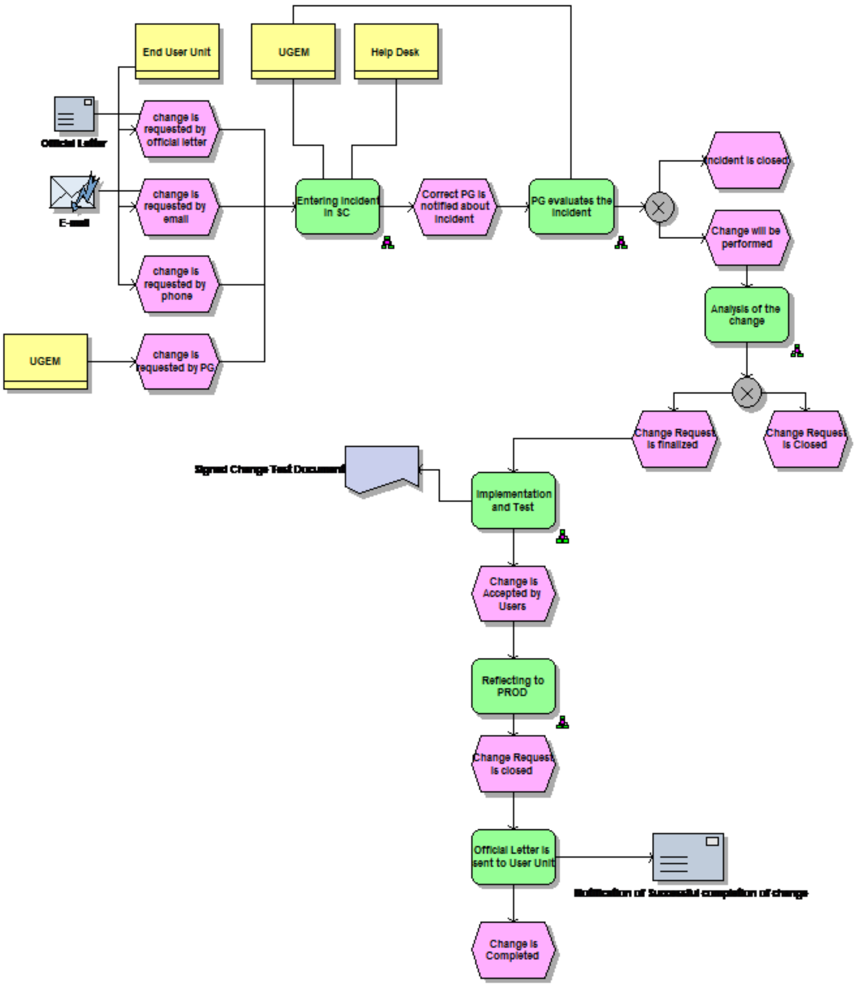

End User Unit
UGEM
Help Desk
Official Letter
change is requested by official letter
E-mail
change is requested by email
change is requested by phone
UGEM
change is requested by PG
Entering incident in SC
Correct PG is notified about incident
PG evaluates the incident
Incident is closed
Change will be performed
Analysis of the change
Change Request is finalized
Change Request is Closed
Signed Change Test Document
Implementation and Test
Change is Accepted by Users
Reflecting to PROD
Change Request is closed
Official Letter is sent to User Unit
Notification of Successful completion of change
Change is Completed

Figure A.1: Case1 - Business Model with eEPC Notation

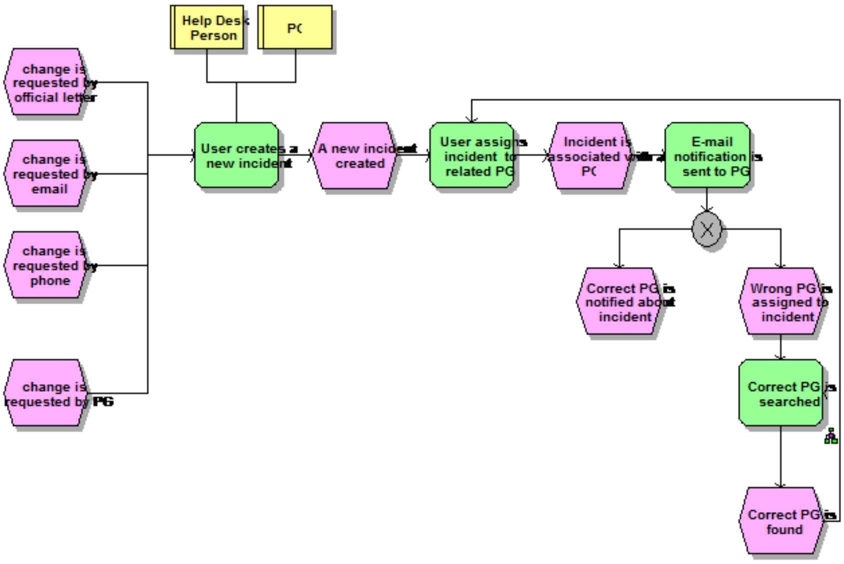

Help Desk Person
PC
change is requested by official letter
change is requested by email
change is requested by phone
change is requested by PG
User creates a new incident
A new incident created
User assigns incident to related PG
Incident is associated with a PC
E-mail notification is sent to PG
Correct PG is notified about incident
Wrong PG is assigned to incident
Correct PG is searched
Correct PG is found

Figure A.2: Case1 - Business Model with eEPC Notation - Enter Incident

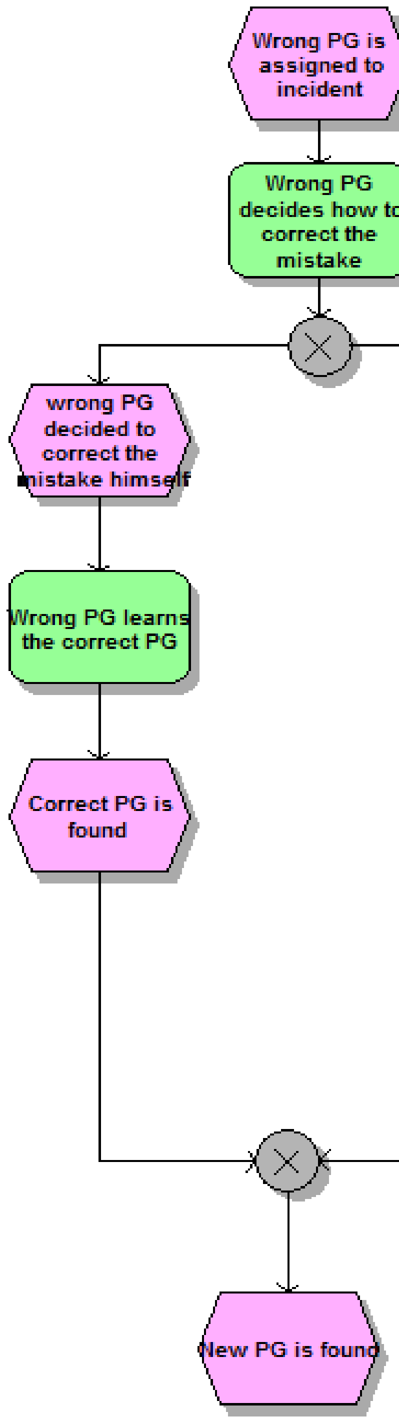

Wrong PG is assigned to incident
Wrong PG decides how to correct the mistake
wrong PG decided to correct the mistake himself
Wrong PG decided to inform HD about the mistake
Wrong PG learns the correct PG
PG warns Help Desk about the wrong assignment
Help Desk learns the correct PG
Correct PG is found
Correct PG is found
New PG is found

Figure A.3: Case1 - Business Model with eEPC Notation - Find PG

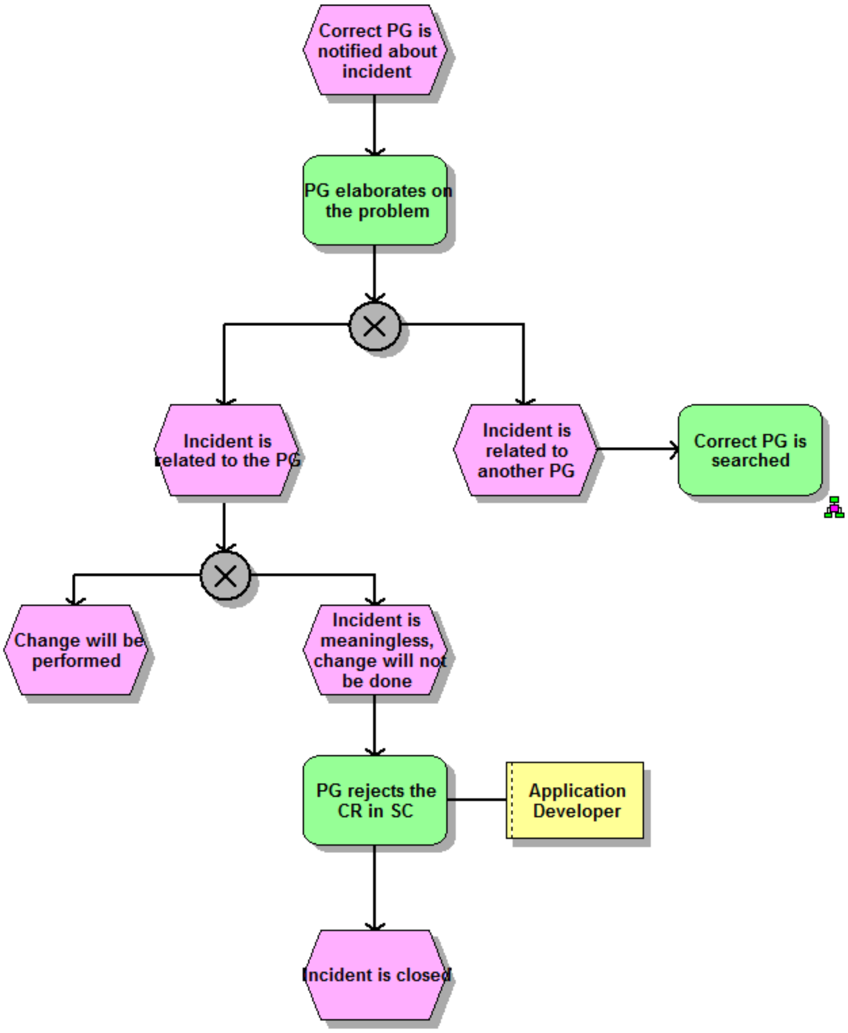

Correct PG is notified about incident
PG elaborates on the problem
Incident is related to the PG
Incident is related to another PG
Correct PG is searched
Change will be performed
Incident is meaningless, change will not be done
PG rejects the CR in SC
Application Developer
Incident is closed

Figure A.4: Case1 - Business Model with eEPC Notation - Evaluate Incident

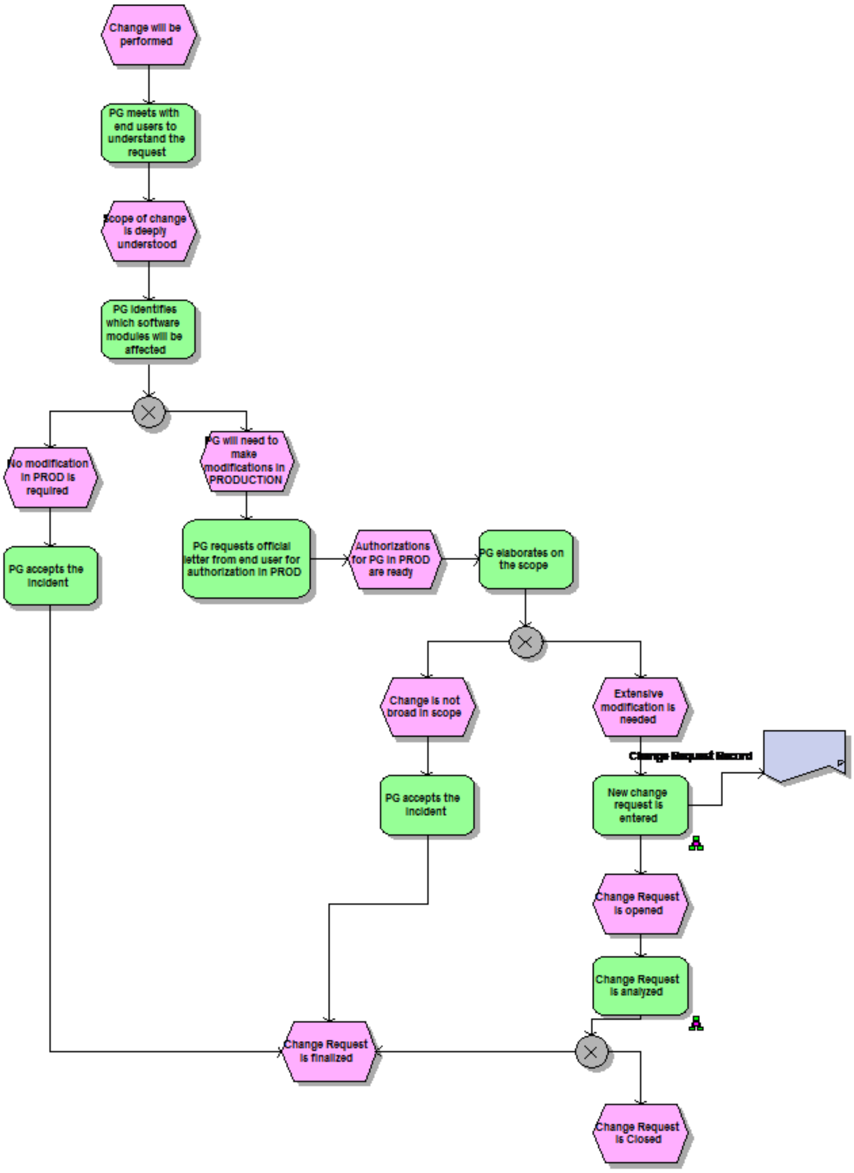

Change will be performed
PG meets with end users to understand the request
Scope of change is deeply understood
PG identifies which software modules will be affected
No modification in PROD is required
PG will need to make modifications in PRODUCTION
PG accepts the incident
PG requests official letter from end user for authorization in PROD
Authorizations for PG in PROD are ready
PG elaborates on the scope
Change is not broad in scope
Extensive modification is needed
Change Request Record
PG accepts the incident
New change request is entered
Change Request is opened
Change Request is analyzed
Change Request is finalized
Change Request is Closed

Figure A.5: Case1 - Business Model with eEPC Notation - Analysis

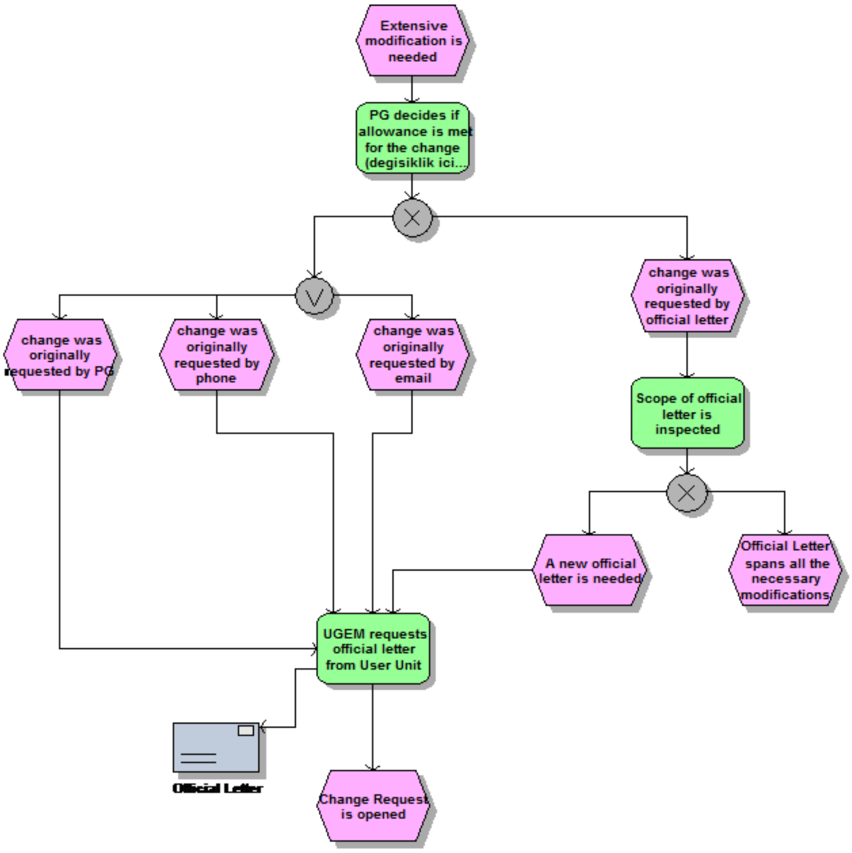

Extensive modification is needed
PG decides if allowance is met for the change (degisiklik ici...
change was originally requested by official letter
change was originally requested by PG
change was originally requested by phone
change was originally requested by email
Scope of official letter is inspected
A new official letter is needed
Official Letter spans all the necessary modifications
UGEM requests official letter from User Unit
Official Letter
Change Request is opened

Figure A.6: Case1 - Business Model with eEPC Notation - New Change

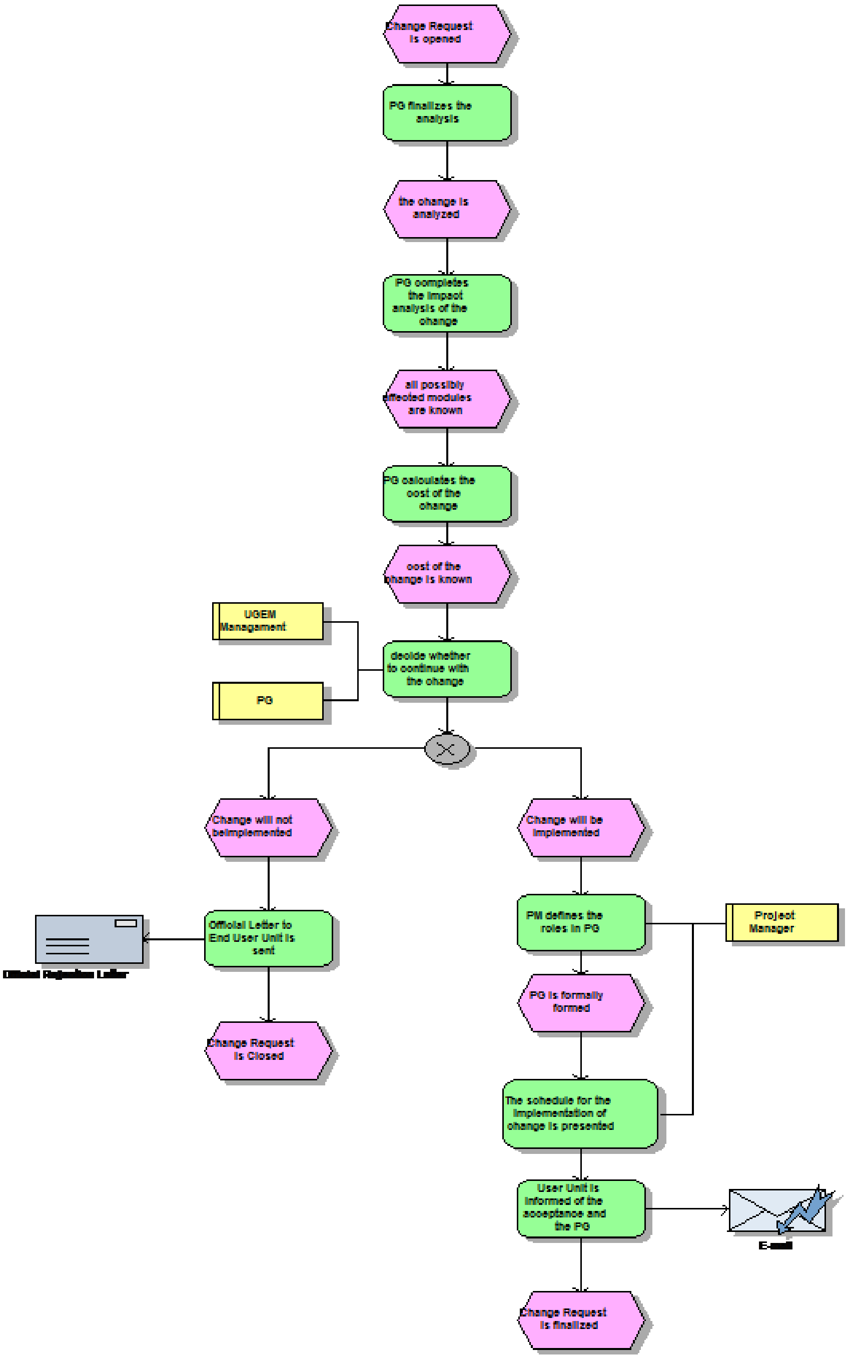


Figure A.7: Case1 - Business Model with eEPC Notation - New Change Analysis

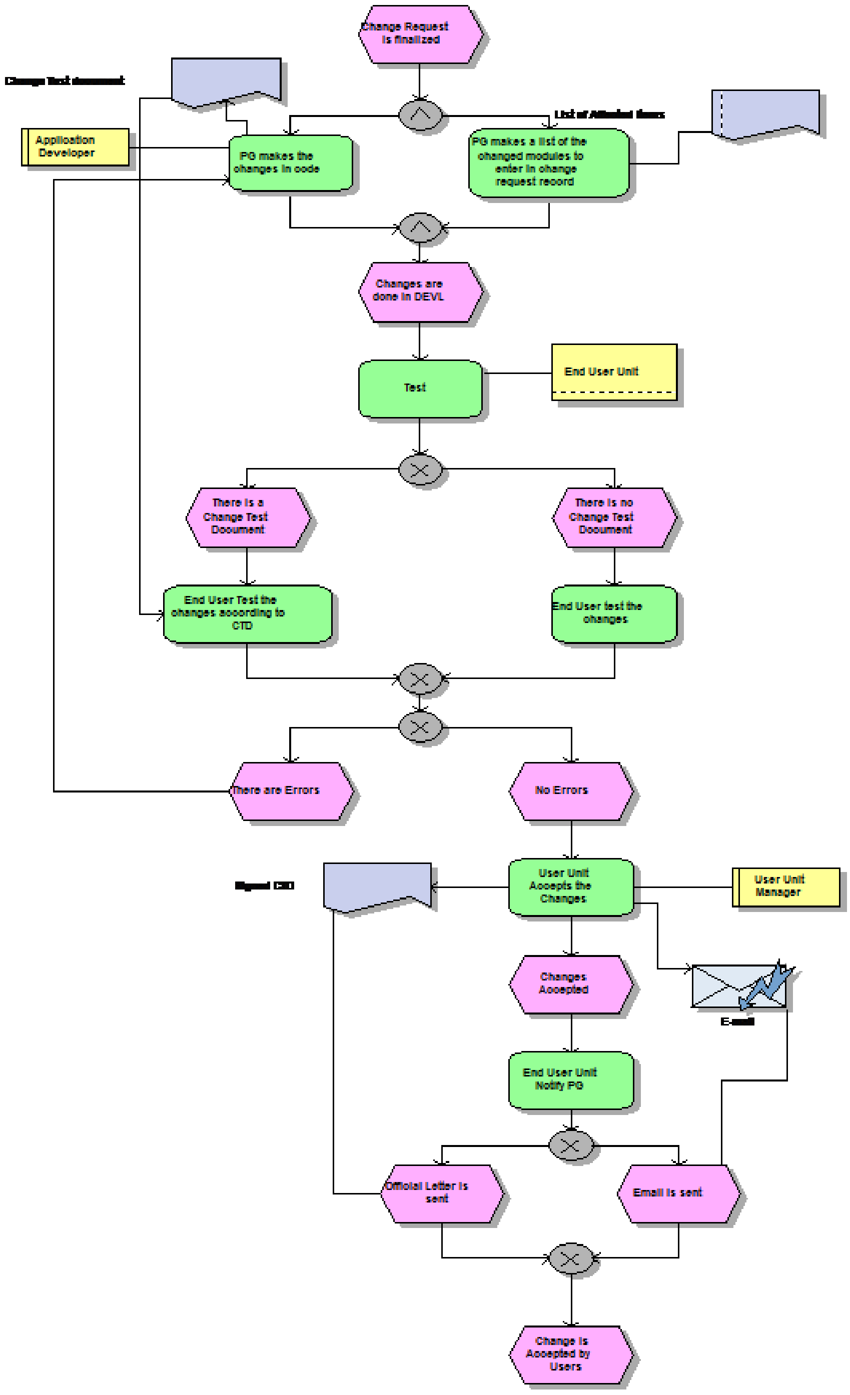

Change Request Is finalized
Change Test document
Application Developer
PG makes the changes in code
List of Affected Items
PG makes a list of the changed modules to enter in change request record
Changes are done in DEVL
Test
End User Unit
There is a Change Test Document
There is no Change Test Document
End User Test the changes according to CTD
End User test the changes
There are Errors
No Errors
Signed CTD
User Unit Accepts the Changes
User Unit Manager
Changes Accepted
E-mail
End User Unit Notify PG
Official Letter is sent
Email is sent
Change is Accepted by Users

Figure A.8: Case1 - Business Model with eEPC Notation - Implementation

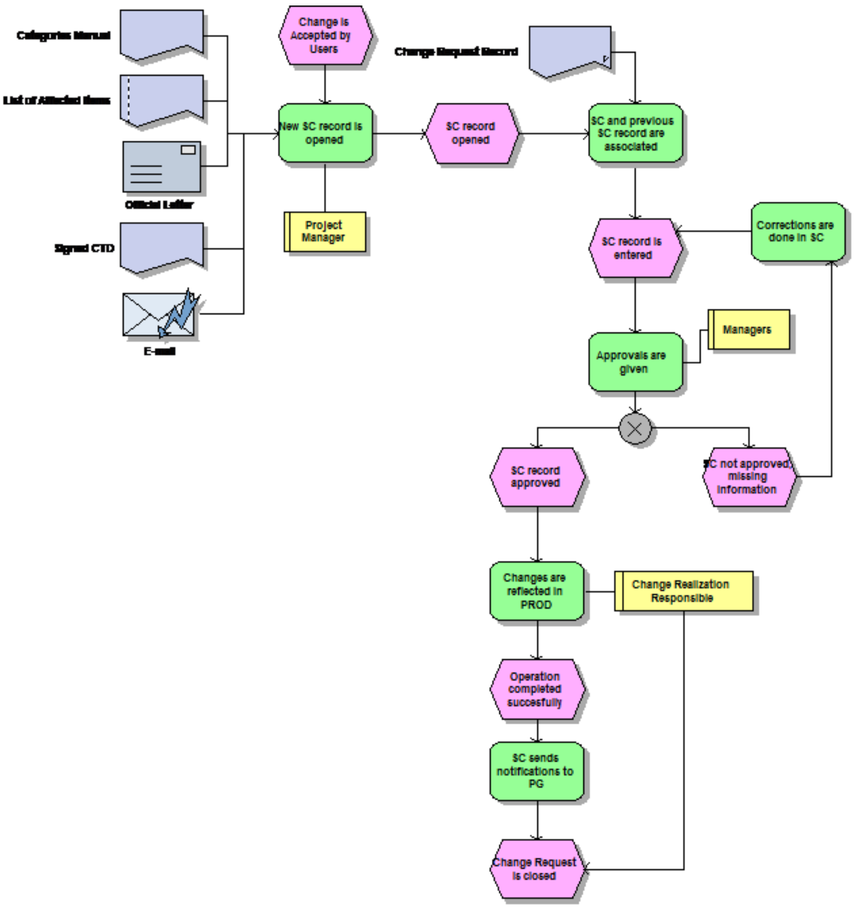

Categories Manual
List of Affected Items
Official Letter
Signed CTD
E-mail
Change is Accepted by Users
New SC record is opened
Project Manager
SC record opened
Change Request Record
SC and previous SC record are associated
SC record is entered
Corrections are done in SC
Approvals are given
Managers
SC record approved
SC not approved, missing information
Changes are reflected in PROD
Change Realization Responsible
Operation completed succesfully
SC sends notifications to PG
Change Request is closed

Figure A.9: Case1 - Business Model with eEPC Notation - Deployment

Deploying processes requires specific notations for each process engine:

- Sample XML Schema and WSDL Definitions to deploy to Intalio engine can be found at http://processexecution.wordpress.com/
- Sample BPMN20.xml format for deployment to Activiti engine can be found at http://processexecution.wordpress.com/

CURRICULUM VITAE

Makbule Filiz Çelik Yeşildoruk was born in Sinop, Turkey in 1978. She received her bachelor degree in Computer Engineering Department in Bilkent University in 2000. In 2006, she had her M.S degree in Information Systems in the Informatics Institute of Middle East Technical University. She has been working as a software engineer since 2000.

Her research interests include; process improvement, business process modeling and business process management

Publications

1. F. Celik and O. Demirors. Process Modeling Methodologies for Improvement and Automation. Proceedings of 2011 IEEE International Conference on Quality and Reliability.2011

2. F. C. Yesildoruk and O. Demirors. Utilizing Process Definitions for Process Automation: A Comparative Study. BPM and Workflow Handbook. 2010.

3. F. Celik, B. Aysolmaz and O. Demirors. The Tool Coverage of Software Process Improvement Frameworks for Small and Medium Sized Enterprises. Product-Focused Software Process Improvement 10th International Conference. 2009.

4. F. Celik, B. Aysolmaz and O. Demirors. Kucuk ve Orta Olcekli Isletmeler Icin Gelistirilen Surec Iyilestirme Cercevelerinin Kapsami. UYMS'10. 2010.