

# Modeling Business Processes to Generate Artifacts for Software Development: A Methodology

Banu Aysolmaz  
Middle East Technical University  
Informatics Institute  
Ankara, Turkey  
banu@aysolmaz.com

Onur Demirors  
Middle East Technical University  
Informatics Institute  
Ankara, Turkey  
demirors@metu.edu.tr

## ABSTRACT

When business processes are to be automated by an information system, business process knowledge is required in many ways throughout the software development life cycle (SDLC). Frequently, analysis is repeated to recapture this knowledge resulting in duplicate effort and inconsistent artifacts. We present our unified business process modeling methodology, UPROM that is used to generate various artifacts from business process models developed in conformance with its notation and approach. Together with business process models, the artifacts are user requirements document, software size estimation, process metrics list, process definition document and business glossary. These artifacts are easily maintainable as they originate from a single source and generated automatically by a tool. The methodology is applied in three case studies and summary of results is provided.

## Categories and Subject Descriptors

D.2.2 [Software Engineering]: Requirements/Specifications – *elicitation methods*. D.2.8 [Software Engineering]: Metrics – *Process metrics*. D.2.9 [Software Engineering]: management – *Cost estimation, Time estimation*. H.4.1 [Information Systems Applications]: Office Automation – *workflow management*. K.6.3 [Management of Computing and Information Systems]: Software Management – *Software process*

## General Terms

Documentation, Management

## Keywords

Business process modeling, requirements analysis, COSMIC, software size estimation, business glossary, process metrics, process definition document

## 1. INTRODUCTION

Business process modeling (BPM) is frequently used for transferring process experience into structured process

knowledge. BPM methodologies are useful in defining existing business processes for better understanding and analyzing to discover current problems as well as depicting the to-be processes in alignment with the organization's strategic objectives. The increase in process awareness led to the increase in the development of business application systems to automate business processes of the organizations. These systems can be generally named as process aware information systems (PAIS) [12].

Business process automation covers analysis, implementation and monitoring of processes. This automation can be achieved by workflow management systems built upon high level process automation infrastructures, or any other software system developed by lower level or general purpose coding languages. In any case, it is critical to analyze, reveal and capture the knowledge of business processes in the business domain and carry this information to the technical domain through software development phases in a complete and consistent way.

Business process knowledge is necessary throughout SDLC of systems developed to automate business processes. This knowledge may be explicitly defined as process documentation or models, or it may be inherent in the culture of the organization. By means of BPM activities an organization creates the opportunity to improve its processes while capturing valuable knowledge that can be utilized for other purposes. If the organization can transfer this knowledge to other activities in a systematic way, related artifacts can be developed in a consistent and maintainable way.

In our studies, we identified practices in SDLC that can potentially utilize business process knowledge directly. These are requirements analysis, software size estimation and process documentation. In this paper, we introduce a unified BPM methodology, UPROM, to conduct business process and other analysis activities in an integrated way to develop a set of models. When we develop models by applying UPROM, we can automatically generate the artifacts to be produced as outputs of the above practices. These artifacts are user requirements document, functional size estimation of the PAIS, and process documentation including process definition document, business glossary and process metrics list.

UPROM provides the notation, rules and guidelines to follow during modeling activities so that all the required information is embedded into one source: the models. This ensures complete and consistent artifacts as they all originate from business process models. In addition, maintainability of the artifacts is enhanced. As change is inherent in software systems, interrelated changes between business process models and other software related artifacts need to be tracked [18]. Impact analysis of changes can

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

MiSE'14, June 2 – June 3, 2014, Hyderabad, India  
Copyright 2014 ACM 978-1-4503-2849-4/14/06...\$15.00

be conducted on the single source, and all artifacts can easily be traced back to the models and regenerated.

Business process models are used by business analysts and software engineers in many ways to gather requirements in software development [12, 17, 18, 23]. Business process models are already defined as a means for “documenting, analyzing, improving, and ultimately codifying a set of business process requirements” [15]. But usually, they are not sufficient to define user requirements in a structured and traceable way to explain how the automated system is supposed to behave to conduct each activity in business processes [27]. As part of UPROM, we introduce a method to conduct requirements analysis on business process models to express user requirements either in the form of diagrams or generated as requirements document.

Monitoring the execution of business processes is as important as automation of them. Process metrics, which are the metrics that require business data [20], focus on evaluation of process performance rather than the systems. Process metrics are defined on business process models at business level analysis. This definition is hardly utilized in the technological side as it does not specify the collection of the related data. UPROM provides an approach for modeling process metrics on process models so that definition and collection procedures for process metrics can be generated as a list that can be directly used in development.

Software functional size estimation is essential for time and cost estimation in project planning [25]. For functional size measurement of a software system, a complete set of functional requirements is necessary which specifies data movements for each functional process [1]. This is characteristically achieved when software requirements specifications are developed, which means reliable time and cost estimation can only be conducted after this phase. Requirements defined by UPROM provides valuable information for size estimation before detailed software specification phase, as they provide structured analysis for each function in processes. UPROM incorporates certain procedures to estimate the functional size from business process and analysis models, enabling early size estimation within the SDLC.

One of the aims of process modeling is to ensure consistent usage of terminology throughout the organization. Business glossary covering terminology of the domain is used for this purpose [6]. Achieving an agreed set of definitions is difficult, especially when processes are defined in a decentralized way [29]. In UPROM, data definitions are embedded inside business process models. Business glossary is generated using these definitions. It lists all terminology and their unique definitions throughout models.

Stakeholders need process definition documents to examine the processes in a different perspective, to add detailed information to the processes or both. UPROM specifies procedures to embed metadata and descriptions on the models and generate process documents automatically; thus achieving the two purposes. In this way, both business glossary and process definition document are made available to the stakeholders to be used during analysis of the system and operation phase when the stakeholders use the system to execute their processes.

The summary of all artifacts generated automatically by UPROM and their related phases in SDLC are shown in Figure 1. We utilize UPROM BPM tool to develop models and generate artifacts automatically. This tool is developed based on bflow Toolbox which supports Event Driven Process Chain (EPC)

modeling [21]. As all of these artifacts are generated from models, when a change is required related to any of the artifacts, its impacts are analyzed and updates are conducted on the models. Then change is automatically reflected to all artifacts as they are regenerated.

This paper provides a brief overview of the methodology and summarizes the results of the applications. In section 2, summary of the related research is provided. Section 3 describes UPROM – the unified BPM methodology with the brief introduction to the notation, procedures and artifact generation principals. Section 4 summarizes the results and lessons learned from the case studies. Section 5 presents the conclusion.

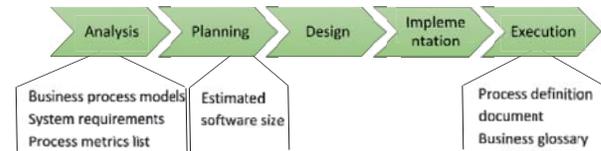


Figure 1. Unified methodology artifacts by SDLC Phases

## 2. RELATED WORK

In this section, we focus on the related work for the usage of business process models in the practices of concern for UPROM.

Requirements are the hardest artifacts to materialize, while they are also the ones most effective on the success of the project [5, 26]. BPM is accepted as a good method for requirements engineering, while the results of the studies show that notations are not expressive enough [3, 10, 26]. Nicolas and Toval’s literature study reveals that no approaches exist to generate natural language statements from business process models and existing approaches do not maintain the link between the generated requirements and initial models [27]. Cox et.al. proposed an approach for deriving software requirements from business process models through the use of problem frames approach [9]. Mayr, Kop and Esberger [23] developed a simple language to conduct requirements modeling during BPM. Borgida et.al. utilized goal models to use as requirements [5]. These studies focus on early requirements and are based on commonly accepted BPM notations. Chiao et.al. developed a data centric approach to determine software requirements [7] which served the purpose of automating process models. Earlier studies of our research group are also examples in this domain providing promising results and establishes the basis of these studies [8, 11].

There are a number of attempts to automate the measurement procedure of the FSM methods by means of software models. Many studies utilize UML models [4, 16, 22] where they do not include the use of business process model. Monsalve et.al. proposed a method to measure software functional size from process models [25]. However it identifies data movements directly on process models, which requires a size estimation expert to conduct the analysis. Our previous studies also provided fruitful results on size estimation using process models [19].

Many international standards and reference models require process documentation and define its contents [30–33]. The process definition document is seen as essential by many stakeholders either together with models or alone. Some BPM tools also have capabilities to generate documentation like in [13] where process elements are listed without any grouping and cross process relations are not considered. Business glossary or data

dictionary is required when business analysis is conducted, which contains business terminology and their definitions [6].

Definition and collection of process metrics during process execution is studied in the academy and industry [20]. Defining metrics to measure the success of processes is an integral part of business analysis, and how those metrics will be used needs to be clarified in this phase [6]. In BPM notations, metrics are defined in abstract level, without details for measuring [28].

In general, it is hard to find BPM approaches that focus on more than one practice to utilize BPM knowledge. Maintainability and traceability of business process models with related artifacts is a problem and generally conducted manually [7].

### 3. UNIFIED BPM METHODOLOGY

When artifacts that can benefit from business process knowledge are developed independently, excessive effort is spent to analyze and define them. This also results in inconsistencies, incompleteness and difficulties to maintain and trace the artifacts. UPROM aims to prevent these problems by transferring business process knowledge to other artifacts in SDLC, and reduce the total effort as redundant data collection and analysis activities are prevented. The following sections describe the approach, modeling notation and generation procedures for the artifacts.

#### 3.1 BPM Diagrams and Generation of Process Documentation

##### 3.1.1 BPM Diagrams

The notation consists of 6 diagram types, all of which are interrelated with a common *metamodel*. The notation is based on eEPC [28], supported with three additional diagram types to model different perspectives of business processes. The following diagram types are used to conduct initial BPM activities.

- **Value Chain (VC) Diagram:** VC Diagram provides a high level view of the activities that add value to the products/services of the organization. It is placed in higher levels of models. Other VC or EPC diagrams can be connected to it as subdiagrams. VC diagram can be a subdiagram only for another VC diagram.
- **Function Tree (FT) Diagram:** FT Diagram shows a number of activities in a hierarchical way. It is used to depict activities for which predecessor-successor relationship does not exist, thus cannot be modeled by EPC diagram. Its subdiagrams can be FT or EPC types and it is placed at higher levels as well.
- **Extended Event Driven Process Chain (EPC) Diagram:** EPC is a well-known diagram type to model process flow using events and functions together with the symbols representing other process perspectives. We utilize this notation as is widely used in BPM field [14] enriching with extra symbols and metamodel definition. EPC can have another EPC diagram under a process interface symbol or another EPC or FAD (which is explained below) under a function linked as subdiagram.
- **Organization Chart (OC) Diagram:** OC Diagram is used to organize all organizational elements utilized in business process models in one diagram and show their relationship. Different types of organizational elements can be defined.

The set of models is organized as a *modeling project*. All diagrams of a modeling project are associated with each other with subdiagram and process module relations, and also *unique*

*object* definitions. The principals to be followed to develop and structure business process models are described below.

Decomposing concepts into levels and depicting information in hierarchical forms is a common way to decrease complexity [5]. To gain a complete view of models and enhance their understandability, process diagrams in UPROM are organized in a hierarchical structure. The set of interrelated diagrams is referred to as a modeling project. Hierarchy is established by inserting a subfolder for a subdiagram, thus increasing the level with each subdiagram. A single process map diagram, which may be a VC, FT or EPC, is placed on the top level of the modeling project hierarchy. Each diagram of VC, FT and EPC types is associated to the diagram on the higher level with a subdiagram relation.

Each process model diagram encapsulated in a folder at any level can be treated as a process module which can be called by another process like a function call in a software code. Within the hierarchy, a process module can be called at any point in the flow of an EPC diagram by means of a process interface symbol.

Usage of multiple start and end events are not supported by workflow nets due to soundness concerns [24]. Multiple start and end events are supported by UPROM to enhance the models' expressiveness of representing alternative scenarios. These events are analyzed to comply with events in other processes so that the flow between the processes can be tracked by means of events.

Another factor that makes modeling project a cohesive structure is the usage of *unique objects* throughout the project. Objects from the same group of symbols which are assigned the same name are maintained as a unique object. For example "order" document in one EPC, "order" file in another EPC and "order" entity in an ER diagram are all the same objects. Altering a property of one object causes a change in all instances of the unique object.

To achieve an integrated set of information by means of models, diagrams and objects are enriched with properties. For each diagram, *metadata* like version, author, scope, description can be assigned. *Description* can be attached to each object, where the user can provide detailed natural language explanation. Both the model properties and descriptions related to functions are then used in the generation of process definition documents.

##### 3.1.2 Generation of Process Documentation

The process definition document presents natural language definitions of processes in a predefined template format. Covering content descriptions from standards [30–33] and example process definition documents from our experiences in many organizations [2], a template is prepared that covers the following headings:

Process data and description (aim, scope, status, version), list of process responsibilities and types, list of inputs, list of outputs, list of entrance and exit criteria (and how these criteria are related to other processes), activities presented in order (including its responsibilities, related application system, inputs, outputs, business rules, subdiagram and detailed description), list of business rules, list of other processes that utilize this process, list of subprocesses and external processes within this process and list of metrics.

The process definition document is automatically generated by UPROM tool having a separate section for each process conforming to this template and uniting them in one document.

Process metrics are modeled on EPC diagrams. A metric is modeled by connecting metric symbol to a function or event to

identify when the metric will be measured, and connecting to an information carrier symbol to designate its sources of information. If one cannot define the metric in this way, this means either processes are not defined thoroughly or the metric is not related to the business process. Ensuring alignment between process definitions and process metrics, a list of process metrics with related information can be obtained from the tool.

Another property attached to specific object types is the *technical term*. It can be defined for organizational elements, applications, information carriers and entities. By means of the uniqueness of objects, it is assured that an object has only one definition through modeling project. Using technical term property, a business glossary is automatically generated covering definition of all technical terms in the processes. This includes organizational elements, applications and other entities with decomposition depicted as modeled on ER diagram.

### 3.2 Analysis Diagrams and Generation of Requirements and Size Estimation

#### 3.2.1 Analysis Diagrams

To conduct user requirements analysis of the system that automates the modeled business processes, each function on business processes are further analyzed by means of two diagram types. The two diagrams supporting the analysis of the system, thus enabling the generation of requirement statements document and software size estimation are explained below.

- **Function Allocation (FA) Diagram (or FAD):** FA diagrams are modeled for each function on EPC models which are envisioned to be automated by the system to be developed. They are placed as a subdiagram for the related function and become the leaves in the process model hierarchy, thus no other subdiagram can be attached to them.

We expect FA diagrams to answer three aspects of the system to identify how the related function is to be executed:

- Who has the responsibility to conduct the function and what type of participation she has (in RASCI format)?
- What type of entities is manipulated during the conduct of the function with what operation?
- What are the constraints to be considered by the system as the function is executed?

FA diagram is designed to answer these questions while conducting business level analysis with end users. The objective is not to determine the most accurate way on how the software works, but to reveal the needs of the system from the user perspective. FAD then will be used as an input to specify software requirements specifications in further phases.

The function for which FAD is created is placed in the center of the diagram and the rest of the objects are organized around it. A sample FAD from a case study can be seen in Figure 2. FAD is designed to include the following elements and connections:

- **Organizational Element – Function:** This connection is established to answer first question. Roles rather than organizational units are used to specifically identify the responsibility. 5 connection types can be utilized to express RASCI information. “Carries out” is used on Figure 2.

- **Function – Entity:** This connection specifies the entities that are manipulated by the function based on CRUDL operations. Possible connection types are: “uses, views, creates, changes, reads, deletes, lists”. Five entities are placed on Figure 2 connected with types of “views, uses, creates, changes”.
- **Entity – Application:** A function may utilize and manipulate entities from more than one application. Each entity is connected to an application to specify the application on which the related entity resides. Three of the entities in Figure 2 resides on one application, and two of them on the other.
- **Application – Constraint:** The rules and constraints that need to be followed by the application during the execution are depicted. The constraints can be about state changes, limitations in values, constraints between entities and so on. Three constraint objects in green color are placed in Figure 2.

- **(Conceptual) Entity Relationship (ER) Diagram:** ER Diagram aims to compose a conceptual and holistic view of the entities in the system identified during FAD modeling and their relationship between each other. ER diagram here is not an attempt to provide a fully structured E-R database model on which software development can be based. Instead, it is a conceptual schema of what entities need to be processed during implementation of process models on which business users can come to an agreement. Entity and cluster decompositions are modeled by aggregation relation. Attributes can be modeled by attribute symbols connected to entities. Generalization and relationship symbols are utilized to depict specific relations between entities and clusters.

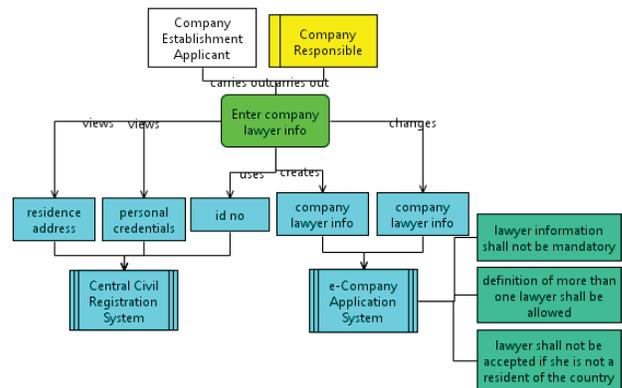


Figure 2. A view of an example Function Allocation Diagram

#### 3.2.2 Generation of Requirements Statements

FA Diagram is used as a tool to express user level requirements of the system in UPROM. The diagrams can be utilized as requirements in the form they are modeled. However, to enhance their readability and understandability by a wider group of stakeholders, natural language requirement statements are generated and formed into a requirements document by UPROM tool. Three types of sentences are generated as explained below. The statements in the quotations and in italics are filled with the names of related objects from the model.

- Type 1 – Sentence for specifying responsibilities

“*Responsibility 1*” shall “*carry out*”, “*responsibility 2*” shall “*approve*”, “*responsibility 3*” shall “*support*” (...) the operation of “*function 1*”.

- Type 2 – Sentence for expressing data manipulations

During “function 1” by using “entity 1”, “entity 2” shall be “viewed”, “entity 3” shall be “created” (...) on the system “Application 1”.

- Type 3 – Sentence for constraints

During “function 1” on “application 1” system, “constraint 1”.

Sentences are extended when there are multiple organizational elements and applications on FAD for type 1 and 2 sentences. Separate sentence is formed for each constraint conforming to type 3 sentence form. The study is conducted in Turkish language and the expressions are completed as required by language rules.

### 3.2.3 Generation of Software Size Estimation

COSMIC functional size measurement method states that “a piece of software interacts with its functional users via data movements across a boundary” [1]. By using business process models and FADs, we can identify functional users and functional processes of the system to be automated in system level. There are four data movement (DM) types: Entry, Read, Write, eXit. CRUDL operations modeled on FADs capture information on data manipulations, although at this phase it is not as accurate as identifying them from software specifications. Thus, the operations can be utilized to make an *early estimation* of total software size, which can be used for planning purposes before the project goes through detailed software requirements analysis.

**Table 1. Operations on entities and corresponding DMs**

Operation on Entity	Related DM
Create, Delete	E, W
Change	E, R, W, X
View, List	E, R, X
Use, Read	R

To determine the size of an application introduced by each FAD, the number of DMs for each operation on the entities is calculated using Table 1. Further rules are introduced to enhance the accuracy of the measured size. ER diagram is an important source for such rules. For example, additional DMs of Entry type are removed when entities on a FAD are related to each other by relationships and aggregations. A size estimation report is generated automatically conforming to these rules. DMs for each FAD and total sizes in Function Points (FP) for each application placed on the system can be found in this report.

## 4. CASE STUDIES AND RESULTS

UPROM methodology is used in three cases. Case study research is applied for two systems within an e-government program: Company and Trademark Central Registration Systems. The aim of the projects was to analyze processes and develop technical contract documents for software development tender. UPROM was followed thoroughly to prepare the deliverables. Data is collected as metrics on artifacts, observations and interviews. Subject matter experts (SMEs) and analysts analyzed processes by developing UPROM diagrams. 21 EPC and 118 FA diagrams were developed with 150 function objects. Interviews are conducted with 7 analysts and SMEs. All interviewees agreed that process definition document and business glossary with 120 entries helped to enhance understandability and was useful for the analysis of the processes. They also stated that UPROM enabled

them to achieve a complete and consistent set of requirements. Generated 540 requirements constituted 90% of the statements in the technical contract. A measurement expert manually measured a process and compared results to estimated size. When the size deviation for each function is summed up, deviation was 25%. When the size is added up for the whole process, it deviated by 6%. Estimated software size is utilized to manage the tender.

The second case study was performed for a set of three small business application systems to evaluate the size estimation capabilities of UPROM. Business process and analysis models were developed for these systems using existing requirements document. Generated size estimation results are compared to actual software sizes. Functional size is measured 53 FPs and the estimated size deviated by 3.6% on average for the three systems.

The third case study was performed on the Public Investment Processes of Ministry of Development. The project scope consisted of development of business process models, process definition document and data dictionary. 10 EPC and FT diagrams were developed with 65 function objects. Business glossary had 93 entries. SMEs, which were initially not BPM literate, emphasized that they prefer to use models and process definition document together and require BPM in their oncoming studies. Four analysts that were not involved in UPROM studies before agreed that UPROM and the tool supported them to analyze and define processes effectively and generate the artifacts in a consistent, complete and maintainable way.

## 5. CONCLUSIONS

This paper introduced a unified BPM methodology, UPROM, to develop business process models in an integrated way and generate various artifacts to be used in SDLC. The methodology is useful when a PAIS is to be developed to automate the business processes. Business process models are defined using EPC, FT, VC and OC diagram types by conforming to the specified rules and guidelines. Requirements analysis is conducted in an integrated way by means of FA and ER diagram types. All diagrams constitute a whole as a modeling project, representing different perspectives of business processes. The modeling project contains all the required information to generate artifacts from a single source. UPROM tool is utilized to generate artifacts of user requirements document, software size estimation report, process definition document, business glossary and process metrics list in conformance to generation procedures. These artifacts can be utilized in related phases of SDLC as shown in Figure 1 and used as input to other phases as the software developers and end users require.

We applied the methodology in three cases. Business process models and artifacts produced by UPROM in these cases are being used by stakeholders in real life settings. Both domain experts and analysts emphasize the ease of maintainability and traceability of the artifacts. In case of any changes, the effects of the change are analyzed and updates are done on the modeling project and all the artifacts are regenerated. Hence, very little maintenance effort is spent on the artifacts in case of changes.

## 6. REFERENCES

[1] Abran, A., Desharnais, J.-M., Oigny, S., St-Pierre, D. and Symons, C. 2009. *The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual*.

- [2] Aysolmaz, B. and Demirörs, O. 2011. A Detailed Software Process Improvement Methodology: BG-SPI. *Systems, Software and Service Process Improvement SE - 9*. R. O'Connor, J. Pries-Heje, and R. Messnarz, eds. Springer Berlin Heidelberg. 97–108.
- [3] Berenbach, B., Paulish, D., Kazmeier, J. and Rudorfer, A. 2009. *Software & Systems Requirements Engineering: In Practice*. McGraw-Hill, Inc.
- [4] Berg, K., Dekkers, T. and Oudshoorn, R. 2005. Functional size measurement applied to UML-based user requirements. *2nd Software Measurement European Forum, SMEF* (2005).
- [5] Borgida, A., Dalpiaz, F., Horkoff, J. and Mylopoulos, J. 2013. Requirements models for design- and runtime: A position paper. *Modeling in Software Engineering (MiSE), 2013 5th International Workshop on* (2013), 62–68.
- [6] Brennan, K. 2009. *A guide to the Business Analysis Body of Knowledge (BABOK guide), version 2.0*. IIBA International Institute of Business Analysis.
- [7] Chiao, C.M., Kunzle, V. and Reichert, M. 2013. Integrated modeling of process- and data-centric software systems with PHILharmonicFlows. *Communicating Business Process and Software Models Quality, Understandability, and Maintainability (CPSM), 2013 IEEE 1st International Workshop on* (2013), 1–10.
- [8] Coskuncay, A., Aysolmaz, B., Demirors, O., Bilen, O. and Dogan, I. 2010. Bridging The Gap Between Business Process Modeling And Software Requirements Analysis: A Case Study. *MCIS 2010 Proceedings*. (2010), Paper 20.
- [9] Cox, K., Phalp, K.T., Bleistein, S.J. and Verner, J.M. Deriving requirements from process models via the problem frames approach. *Inform Software Tech.* 47,5(2005),319-337.
- [10] Davies, I., Green, P., Rosemann, M., Indulska, M. and Gallo, S. 2006. How do practitioners use conceptual modeling in practice? *Data Knowl Eng.* 58, 3 (Sep. 2006), 358–380.
- [11] Demirors, O., Gencel, Ç. and Tarhan, A. 2003. Utilizing business process models for requirements elicitation. *Euromicro Conference, 2003*. (2003), 1–4.
- [12] Dumas, M., Aalst, W. Van der and Hofstede, A. ter 2005. *Process-aware information systems : bridging people and software through process technology*. John Wiley & Sons.
- [13] Eid-Sabbagh, R.-H., Hewelt, M. and Weske, M. 2013. A Tool for Business Process Architecture Analysis. *Service-Oriented Computing SE - 61*. S. Basu, C. Pautasso, L. Zhang, and X. Fu, eds. Springer Berlin Heidelberg. 688–691.
- [14] Fettke, P., Loos, P. and Zwicker, J. 2006. Business Process Reference Models : (2006), 469–483.
- [15] Gullidge, T. 2010. Integrated Business Process and Service Management. *Handbook on Business Process Management 1 SE - 22*. J. Brocke and M. Rosemann, eds. Springer Berlin Heidelberg. 481–496.
- [16] Heričko, M., Rozman, I. and Živkovič, A. 2006. A formal representation of functional size measurement methods. *J Syst Software.* 79, 9 (2006), 1341–1358.
- [17] Indulska, M., Green, P., Recker, J. and Rosemann, M. 2009. Business Process Modeling: Perceived Benefits. *Conceptual Modeling - ER 2009 SE - 34*. A.F. Laender, S. Castano, U. Dayal, F. Casati, and J. Oliveira, eds. Springer Berlin Heidelberg. 458–471.
- [18] Jamshidi, P. and Pahl, C. 2012. Business process and software architecture model co-evolution patterns. *Modeling in Software Engineering (MISE), 2012 ICSE Workshop on* (2012), 91–97.
- [19] Kaya, M. and Demirors, O. 2011. E-Cosmic: A Business Process Model Based Functional Size Estimation Approach. *Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on* (2011), 404–410.
- [20] Kung, P., Hagen, C., Rodel, M. and Seifert, S. 2005. Business process monitoring measurement in a large bank: challenges and selected approaches. *Database and Expert Systems Applications, 2005. Proceedings. Sixteenth International Workshop on* (2005), 955–961.
- [21] Kühne, S., Kern, H., Gruhn, V. and Laue, R. 2010. Business process modeling with continuous validation. *J Softw Maint Evol-R.* 22, 6-7 (2010), 547–566.
- [22] Marín, B., Giachetti, G. and Pastor, O. 2008. Measurement of Functional Size in Conceptual Models: A Survey of Measurement Procedures Based on COSMIC. *Software Process and Product Measurement SE - 15*. R. Dumke, R. Braungarten, G. Büren, A. Abran, and J. Cuadrado-Gallego, eds. Springer Berlin Heidelberg. 170–183.
- [23] Mayr, H.C., Kop, C. and Esberger, D. Business Process Modeling and Requirements Modeling. *Digital Society. ICDS '07. First International Conference on* (2007), 8.
- [24] Mendling, J., Neumann, G. and Aalst, W. 2007. Understanding the Occurrence of Errors in Process Models Based on Metrics. *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS SE - 9*. R. Meersman and Z. Tari, eds. Springer Berlin Heidelberg. 113–130.
- [25] Monsalve, C., Abran, A. and April, A. 2011. Measuring Software Functional Size From Business Process Models. *Int J Softw Eng Know.* 21, 03 (May 2011), 311–338.
- [26] Monsalve, C., April, A. and Abran, A. 2012. On the expressiveness of business process modeling notations for software requirements elicitation. *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society* (2012), 3132–3137.
- [27] Nicolás, J. and Toval, A. 2009. On the generation of requirements specifications from software engineering models: A systematic literature review. *Inform Software Tech.* 51, 9 (2009), 1291–1307.
- [28] Scheer, A.-W.W. 1998. *Aris-Business Process Frameworks*. Springer-Verlag New York, Inc.
- [29] Turetken, O. and Demirors, O. 2008. Process modeling by process owners: A decentralized approach. *Software Process: Improvement and Practice.* 13, 1 (2008), 75–87.
- [30] 2010. *CMMI® for Development, Version 1.3 CMMI-DEV, V1.3*.
- [31] 2010. *Control Objectives for Information and Related Technology (COBIT)*. ISACA.
- [32] 2011. *Information Technology Infrastructure Library (ITIL)*.
- [33] 2008. *ISO 9001:2008*.