

# Requirement Elicitation Using Business Process Models

Sander Valvas<sup>(✉)</sup> and Fredrik Milani<sup>(✉)</sup>

University of Tartu, Tartu, Estonia  
sander.valvas@gmail.com, milani@ut.ee

**Abstract.** Oftentimes, when eliciting requirements for system development, the input of domain experts is of great importance. Domain experts are not familiar with artifacts predominantly used by system analysts but rather more attuned to models representing the flow of their work such as business process models. However, these models are rarely used as basis for requirement elicitation. As such, there is a communication barrier, which can cause misunderstandings that translate into imprecise requirements. To address this gap, we propose a systematic approach for eliciting functional requirements using business process models as artifacts in discussion with domain experts. We call this method for Requirement Elicitation from Business Process Models (REB). Based on a mapping of requirement components and elements of business process models, the method guides the discussion between system analysts and domain experts around process models for the purpose of eliciting requirements. The method is validated with a case study and the results show that the REB method was successful in eliciting higher number of relevant requirements with less time required.

**Keywords:** Requirement elicitation · Business process models

## 1 Introduction

One of the initial tasks of requirement elicitation is to uncover and extract needed information to specify a requirement [1]. One main source of valuable information is documentation currently being used by an organization [2]. Such information is commonly captured as written text and as graphical models [3]. Oftentimes, graphical representations of the work being conducted, such as business process models, are used for communicative purposes between various stakeholders, for understanding how work is being performed and where improvements can be made. Business process models, and in particular models following the Business Process Model and Notation (BPMN) language, are widely used as it has been designed to accommodate the needs of domain experts, business analysts and system analysts.

These business process models are valuable sources of information for requirements elicitation. In fact, these models are not only used to understand the environment [4] but increasingly becoming an important part of the requirements specification process [5]. Although there is an abundance of modeling techniques for representing processes, they have their origin in software engineering and therefore, difficult for

domain experts to use. To remedy this gap, extensions of existing software modeling methods (e.g. Activity Diagram in UML) have been developed to also accommodate the needs of the business side. However, business oriented modelers have not adopted such languages and rather prefer to use languages such as BPMN for capturing their business processes as graphical models.

Business process models, while being widely used, are rarely utilized as the main artifact when discussing requirement with domain experts. The lack of a method for systematic elicitation of requirements from business process models might be one reason for such models not being used more. In light of this context, this paper aims at presenting and validating a systematic method for eliciting requirements that are complete (include all the data needed for a requirement), consistent (with no internal contradictions), bounded (include relevant data for the software engineering project) and at the appropriate level of detail. The research question is therefore *“how can requirements be systematically elicited from business process models when engaging with domain experts?”*

The REB method provides a template, which includes the data needed for a requirement, and a set of questions that will guide the elicitation of requirements in collaborative discussions, based on business process models, with the domain experts. For each relevant activity, questions are asked of the domain experts that allows for eliciting the intended requirement. The method is validated with a case study for eliciting requirements for customizing and implementing a standard enterprise system.

The rest of the paper is structured as follows. Section 2 introduces the conceptual foundation of our method and Sect. 3 describes the proposed method. Next, Sect. 4 introduces the case study and the findings. Section 5 discusses related work while Sect. 6 draws conclusions and outlines future work.

## 2 Conceptual Foundation

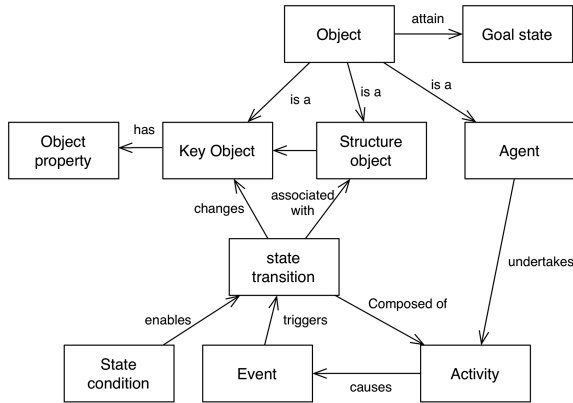
In this section, we first define the necessary components of a requirement and discuss the elements of a business process model. In order to show that necessary components of a requirement are, either explicitly or implicitly, present in business process models, we use BPMN to map the requirement components with business process elements.

### 2.1 Components of a Requirement

Generally, the definition of a requirement is that it describes what a product must do and how it should be done [3]. However, such a definition is general and therefore, many attempts have been made to specify or decompose in more detail the components of a requirement [6]. The decomposition offered by Domain Theory for Requirements Engineering [7] is considered as complete [8] and is widely accepted within the field of requirement engineering [9].

Domain Theory for Requirements Engineering [7] provides a structure of knowledge types (see Fig. 1) present in RE and suggests domain knowledge to be represented in two types of generic models: Object System Models (OSM) and Information System

Models (ISM). OSM describes the essential transaction of the application in terms of a set of cooperating objects and their behavior whereas ISM contains processes that provide information about an OSM.



**Fig. 1.** Meta-schema of knowledge types for domain modeling [7]

The knowledge types that form the basic components of a requirement are “*object*”, “*state transition*”, “*goal state*”, “*activity*”, “*event*”, and “*stative condition*”.

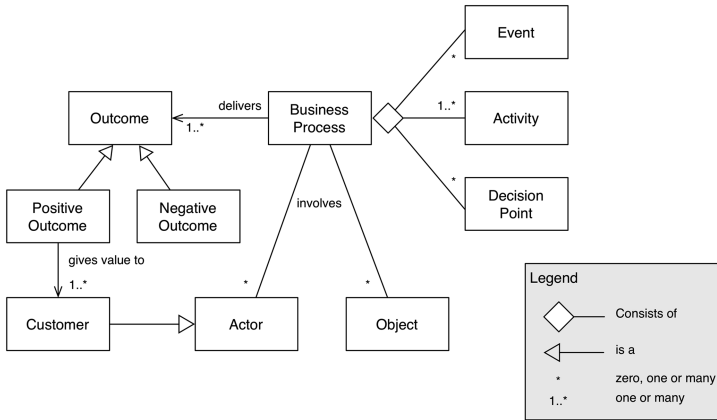
An *object* can be of type “*key object*”, “*structure*” or “*agent*”. A “*key object*” is the subject matter of the essential system transaction. A “*structure object*” is a passive objects that are approximations for real entities, which normally do not appear in data models (e.g. a warehouse or a library). They are therefore persistent, have spatial properties and express containment or possession of “*key objects*” (such as a library contains books). An “*agent*” carries out “*activities*”, which may then create “*events*” initiating “*state transitions*”. A “*state transition*” changes the state of an “*object*” by transferring its membership between “*structure objects*” to achieve a desired “*goal state*” (e.g. a borrowed book moves from the library to the borrower).

A “*goal state*” describes a future, required state, which the system should satisfy, maintain or in some cases, avoid. A goal is specified by describing the state, which is to be achieved, or by describing the algorithms/processes that must be carried out. An “*activity*” is a process, which normally runs to completion and resulting in a state change. An “*activity*” is carried out by actors and triggers the state changes and cause “*events*”. An “*event*” is a single point when something happens.

## 2.2 Business Process Model and Notation

BPMN provides a modeling notation for designing and managing business processes [10]. A business process is a collection of related, structured activities or tasks that produce a specific service or product for a particular customer [11].

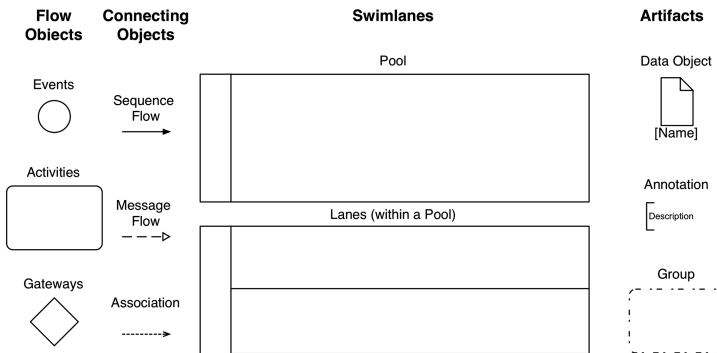
Dumas *et al.* [12] decompose business processes into elements as depicted in Fig. 2.



**Fig. 2.** Ingredients of a business process [12]

An “*event*” represents something that happens and triggers a series of activities. An “*activity*” is a task that takes places for the purpose of fulfilling an operational contract [13]. A “*decision point*” is a point at which a decision is taken that affects the way the process is executed. An “*actor*” is someone or something that performs an “*activity*” or benefits from the output of a process. An “*object*”, be it physical or immaterial, is a thing consumed or produced by an “*activity*”. A process results in “*outcome*” which can be desirable (“*positive outcome*”) or undesirable (“*negative outcome*”).

The above listed elements are represented in BPMN by using a core set of elements (see Fig. 3). These are “*flow objects*” that capture “*events*”, “*activities*” and “*decision points*” (called “*gateways*” in BPMN) as described above. The “*actor*” is represented with “*swimlanes*” that capture “*pools*” and “*lanes*”. The “*artifacts*” represent the “*objects*” and are of either “*data objects*” (data required as in- or output) and “*data store*” (data retrieval or data update) and finally, “*connecting objects*” of BPMN represents the relationship between “*objects*” and “*actors*” with “*activities*”. The core elements of BPMN can be supplemented with different additional markers that specify certain attributes or behaviors of the core elements.



**Fig. 3.** Core set of BPMN elements [14]

### 2.3 Mapping a Requirement to BPMN

Here, we compare the components of requirements to the elements of a business process model. The mapping, as seen in Table 1, shows that all necessary information required for a requirement, can be represented in process models represented in BPMN. In Table 1, the first column lists the components of a requirement; the second shows its corresponding element in BPMN and finally, the third column offers a brief comment.

**Table 1.** Mapping of requirement and BPMN components

Requirement	BPMN	Comment
Key Object	Artifact Events Gateways	Information about a key object can be found in artifacts (data objects, data stores, annotations) or in event types (such as input or output of an event) or in gateways.
Structure Object	Artifact Events Gateways	A structure object is a type of property of a key object and this information in BPMN can be found in the same form as in the case of key objects.
Agent Object	Swim lane (Pool, Lane)	In BPMN, the agent or resources are represented with pools and lanes.
Object Property	Artifact Events Gateways	Key, structure and agent objects have properties that are relevant for requirements definitions. This information can be found in the same form as in the case of key object.
Goal State	Artifact	A goal state is a set of key object properties and is related to structure objects when the process has reached a positive outcome. Since the information about the key and structure objects is found in artifacts, events and gateways, the goal state is also described in the same way as BPMN.
Activity	Activity	Represented as activities in BPMN.
Event	Event	Represented as events in BPMN.
Stative Condition	Artifact, Events and Gateways	Stative condition consists of object properties and therefore, found in artifacts and text annotations of a preceding event or an activity. Also, if the preceding element is a gateway, some of the conditions are described in the outgoing node of the gateway.
Relationship	Connecting objects	Represented as connecting objects in BPMN such as message flows and associations.

As can be seen from Table 1, the components of a requirement, has a corresponding counterpart in business process models as exemplified by BPMN. However, it should be noted that one component of a requirement could have its matching counterpart in several BPMN elements. Nevertheless, the information is either explicitly or implicitly available and therefore, it is possible to elicit functional requirements from business process models.

### 3 Description of the REB Method

In this chapter, we describe the REB method. The method aims at eliciting requirements and as such, its main artifact is a template for requirement specification. The template is filled for every relevant activity of the process model, covering all requirement components as discussed in Sect. 2. The REB method begins with a template, which is populated by asking a set of pre-defined questions from the domain experts while using the business process model as main artifact.

The requirement specification is gradually elicited for each relevant activity. The first step is to determine if the activity is relevant or not. An activity is considered as relevant if it requires some form of functionality support from an information system. If the activity is performed manually and does not require any support from any semi-automated or automated system, it is not relevant. As such, a specification is only elicited for activities that require some sort of support to be considered or developed.

The process of populating the requirement specification is practically achieved by eliciting information about the goal, actor, trigger, operational steps, alternative paths, failure conditions and their management for each relevant activity (see Fig. 4). The information required, is elicited by applying a set of questions that are designed to capture that specific information from the domain experts using business process models as common artifact. As such, the requirement specification template is gradually defined until it forms a complete specification. The process is then repeated for each activity of the business process that is relevant.

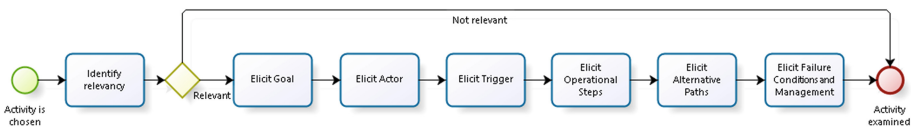


Fig. 4. The REB method.

The REB method requires a set of business process models as input. Access and availability of domain experts are necessary as requirements are being elicited from them by sets of questions related to each step of the elicitation process. The extent, to which the domain experts are engaged, depends on the level of detail in the business process models. If the process models have been modeled in great detail, most of the information is already captured in the models. In such cases, the input of the domain experts is of a more confirmatory nature. However, if the models are not of such

detailed level, the REB method will elicit that “hidden” information from the domain experts. For instance, if the model lacks artifacts, the REB method will, through its questions, inquire about the objects and thus capture the information from the domain experts. As such, the REB method can identify incompleteness’s in business process models and assist in capturing the business processes in greater detail.

Activities are the focal point of the method. Each requirement specification corresponds to at least one activity. The requirement specification (see Table 2) consists of two columns, where the first one states what data is to be captured in the second column of each row. The data required are “*id*” (a unique id for the requirement specification), “*business process*” (name of the process model in which the main activity of the requirement specification belongs to), “*activity*” (the name of the focal activity that is the object of requirement elicitation), “*goal*” (the expected outcome of the activity), “*actor*” (the performer of the activity), “*trigger*” (what initiates the actor to perform the activity), “*procedures of the activity*” (the operational steps taken to perform the activity, both desired steps and alternative steps required when the desired steps cannot be executed), and “*failure conditions and handling*” (cases where the activity cannot be executed or interrupted in its execution and actions to handle the failures). The requirement specification template, which is inspired by use case specification of Cockburn [15], covers all components necessary for a requirement as aligned with their corresponding business process model elements.

**Table 2.** Requirement specification template

<b>Requirement Specification</b>	
ID:	
Business Process:	
Activity:	
Goal:	
Actor:	
Trigger:	
Steps of Activity (positive scenario)	Operational steps:
	Alternative paths:
Failure conditions and management:	

A requirement specification is populated for each activity. However, in some cases, several activities are so connected that they should be treated as one from the perspective of requirement specification. To determine if two or more adjacent activities should be included in one requirement specification, the following questions as inspired by Cockburn [15] are asked: (1) Are the consecutive activities executed by one person, in one place and at the same time? (2) Is it possible or reasonable to have a break between the activities? If the answers to both questions are “yes”, the execution of the activities are tightly connected and there is no reasonable reason to separate them. They should therefore, be treated as one requirement for the system being built.

When eliciting requirements from process models, it is reasonable to assume that the processes are not captured with enough details needed for the elicitation process. If the information is not captured in the model, the REB method will highlight it and leave it to domain experts to add it to the models. The purpose and phase of the project of the requirement specification determines the level of detail that the requirement is specified. In early stages of an IT project, the requirements do not need to be at the same level of detail as in later stages. The REB method allows the elicitors to determine the level of detail by deciding when the template contains enough detail about the requirement. It should be noted that the questions have the purpose of securing the required information but further discussions and follow up questions that enrich the discussion and clarify the requirements further are expected.

*Step 1 – Identify Relevancy of Activity:* The first step is to determine if the activity is relevant. If the activity requires some form of system support, there is a need for having its functional requirements specified. In order to determine the relevancy of the activity, the following questions are asked:

1. Does the execution of the activity require any support from any computer-based system?
2. Is the system under construction to be involved by providing, executing or receiving data during the execution of the activity?
3. Are there any connections to external systems involved in the execution of the activity that need to be considered for interfacing with the system under construction?

*Step 2 – Elicitation of Goal:* An executed activity serves to fulfill a certain predefined outcome or goal. In this step, the outcome of the activity is elicited by asking the following questions:

4. What changes after the activity has been executed?
5. What is required to be achieved or accomplished with the execution of the activity?
6. In what form and/or format are the results in?

*Step 3 – Elicitation of Actor:* In this step, the executor of the activity is elicited. The actor can be either human such as a role, department or organizational unit or a non-human resource such as an information system. The actors are elicited by asking the following question:

7. Who are the actors, human and non-human, who are involved in the execution of the activity?

*Step 4 – Elicitation of Trigger:* Triggers determine when an activity is to be executed. Activities are generally triggered by either an actor receiving a message, a specific predefined time or by the end of a preceding activity. The following questions assist in eliciting the triggers:

8. How does the actor (human or non-human) know when to start the execution of an activity?



9. If it is a message, what kind of message is it and how does the actor become aware of receipt of the message?
10. If the activity is time-dependent event, how is the actor notified about when to start the execution of the activity?
11. If it is complete execution of the preceding activity that is the trigger, is the actor responsible for the execution of the preceding activity and if not, how is the actor informed about it?

*Step 5 – Elicitation of Operational Steps:* An activity usually consists of procedural or operational steps, i.e. the individual steps that need to be carried out in order to execute the activity. In this step, the preferred or desired operational steps are elicited by asking the following questions:

12. What are the operational steps required for the execution of the activity?
13. Who performs the operational steps?
14. What tools or aids does the actor engage or use in carrying out the operational steps (such as human or non-human actors, internal or external systems)?
15. How are these tools or aids used?
16. Are verifications required in carrying out the operational steps?

*Step 6 – Elicitation of Alternative Paths:* Alongside the standard set of operational steps, there are alternative paths taken when the standard cannot be executed. This could be for instance, entering an order when the customer is not registered and an alternative path is required before the order can be registered. These alternative paths are elicited by asking the following questions:

17. Are there cases (when carrying out the standard operation steps) where additional or alternative steps need to be taken in order to reach the goal of the activity?
18. What are the conditions of these cases?
19. What complementary or replacing steps need to take place in such cases?

*Step 7 – Elicitation of Failure Conditions and Failure Management:* Activities cannot always successfully be executed and reach its goal as they might be interrupted or disrupted. In this step, such conditions that hinder an activity from being initiated, interrupted or disrupted are elicited. Furthermore, such failure situations require additional steps to be taken in order to solve the disruption. These failures and steps to manage them are elicited with the aid of the following questions:

20. What can hinder the initiation of an activity?
21. What can cause to interrupt or disrupt an activity?
22. What activities or steps are needed to limit the loss, handle or resolve issues so an activity can be initiated?

## 4 Case Study

The case study method allows researchers to investigate a phenomenon within its real-life context [16], particularly when the boundaries between what is studied and its context are unclear [17]. Case studies are often used for exploratory purposes, but they

are also suitable for testing a hypothesis in a confirmatory study [16, 18] or to evaluate a method within the software and systems engineering domain [19]. These features make the case study method applicable to validate our proposed method.

Our research question is: “*how can requirements be systematically elicited from business process models when engaging with domain experts?*” The purpose of our method is to elicit requirements that are complete (covering the required set of requirements for a system to be constructed) and relevant (requirements that the system being constructed need to have in order to support the business process it was designed to support).

#### 4.1 Setting and Design

The case study is on the quality assurance process of a mid-sized electric engine manufacturing company. The company completed a full implemented an ERP system including process modeling, requirement elicitation, configuration, testing and implementation of the final system. As such, the process models had already been modeled, all requirements elicited and implemented. Furthermore, as the project, at the time of the execution of this case study, had been fully operational for 2 months, all major requirements that had been missed during the elicitation process but implemented in the project were identified. As such, the project had a (1) list of requirements elicited prior to the implementation of the system and (2) a final set of requirements that were implemented including those that were discovered during the implementation of the system. A system analyst, an expert on the ERP system, elicited the requirements of the project. The analyst did not apply any formal method but rather relied on his own expertise and extensive knowledge of the system and expertise from previous implementation of similar projects.

The design of the case study consists of 3 parts. The first part consisted of a series of meetings with the business analyst who headed the implementation project. In this part of the case study, the REB method was introduced followed by meetings where the requirements were elicited with the REB method. The second part was converting the requirements to make them comparable with the two sets of requirements elicited prior to and during the project implementation. The final part of the case study consisted of analysis and comparison of the results.

#### 4.2 Execution

The first author conducted the case study with the lead domain expert of the implementation project. The first part of the case study began with a description of the REB method including the template for requirement specification, its components, some of the questions and an illustrative example. Following the introduction, the elicitation process began. For each activity of the process model, a requirement specification was created. Table 3 illustrates the requirement specification for “*check and update order confirmation*”.

**Table 3.** Example of a populated requirement specification (abridged due to space limitation)

Component	Description
ID:	003
Business Process:	Supply chain security (purchase)
Activity:	Check and update order confirmation
Goal:	Updated order (suggested delivery date and order status updated)
Primary Actor:	Purchase department
Trigger:	Order confirmation received by e-mail
Steps of Activity (positive scenario)	<b>Operational steps:</b>
	<ol style="list-style-type: none"> <li>1. Open PDF format order confirmation received by email</li> <li>2. Find the relevant purchase order</li> <li>3. Check that ordered materials are the same as on the order</li> <li>4. Enter suggested delivery date and change the status to "Confirmed"</li> <li>5. Reply the email confirming the order confirmation</li> <li>6. Save the order</li> </ol>
	<b>Alternative paths:</b>
	<ol style="list-style-type: none"> <li>1. If order confirmation differs from the order (e.g. quantity smaller than ordered), contact the person who created the order and ask for advice; If changes OK follow the normal flow.</li> <li>2. If suggested delivery date is later than the needed delivery date, take same actions as in alternative path 1.</li> </ol>
Failure conditions and management:	<ol style="list-style-type: none"> <li>1. If order confirmation differs from the order and is not acceptable, the order will be deleted and the process will be interrupted.</li> </ol>

The activity, “check and update order confirmation” was determined as relevant (step 1) as it requires some interaction with a system support (not a purely manual task). Then, the goal of the activity was elicited (step 2). With the aid of the questions, it became clear that the goal of this activity is to achieve an updated order. After this, the actor was determined (step 3) which was someone from the purchasing department. The next set of questions aim at eliciting the trigger of the activity (step 4). In this case, a message event preceded the activity indicating that an incoming message from the supplier is the trigger. This was further clarified (with the aid of the questions) that the trigger is an email from the supplier with an attachment. Further discussion revealed that there is no need for any automation or an interface. Next, the operational steps of the activity were elicited (step 5). By using the questions, the operational steps were elicited and clarified. Some steps, such as the second step, “find the relevant purchase order” were elaborated as to what parameters are used to find an order. Following the operational steps, the alternative paths (step 6) were elicited. The discussions based on the questions of the method, revealed that two alternative paths exist, one for when the confirmation differs from the order and when the suggested delivery date is later than the customer needs the goods. The final step of the method (step 7) is eliciting failure conditions and management. In this step, situations that prevent the activity from

starting or that interrupts/disrupts the activity and the measures needed to be taken are discussed. Naturally, the failures are connected with the operational steps and alternative paths. For instance, the alternative path of order confirmation differs from the order and it is not acceptable, the management of this situation is clarified here. In this case, it is to delete the order and the process is interrupted.

## 5 Findings

In order to make the requirements elicited with the REB method (REB set) comparable with the set of requirements elicited (baseline set) prior to the project start and the set after the project was implemented (final set), they were converted. The final set included elicited requirements that had been discarded during the project implementation and requirements that had been missed in the elicitation process prior to project start. The baseline and the final set of requirements were not documented in the same way as the REB set. For instance, an ERP system provides a certain set of functionalities per default and therefore, such requirements were not recorded in the baseline and the final set. These were added, as they are valid requirements in order to make the sets of requirements comparable. The domain expert verified all converted requirements to ensure comparable results.

### 5.1 Comparison

With the implementation of the REB method, a total of 128 versus 121 (baseline set) requirements were elicited. Number of requirements, valuable as it may be, is not enough but needs to be complemented with the quality of the requirements.

The ratio of irrelevant and incorrect requirements to the total number of requirements is used as approximation for the quality of the requirements elicited. The baseline set of requirements had 6 irrelevant or incorrect requirements out of 121 which gives ratio of 4,9 %. The corresponding ratio for the REB method was 5,4 % (7 out of 128).

The REB method resulted in the elicitation of more requirements of equal quality as compared to the baseline set. However, the final set consists of 128 relevant and correct requirements including requirements that had not been elicited prior to the start of the project but identified as relevant during the implementation. The final set included 19 requirements that the baseline had not elicited, thus the ratio of “missed” requirements is (as compared to the final set) 14,8 %. The corresponding ratio for the REB method is 9,4 %. It should be noted that the REB method elicited 4 requirements out of 6 that was discovered during the project implementation and an additional 2 that were not included in the final set but were considered as both relevant and correct. These have been put on the list of enhancements to be developed.

The baseline set of requirements took 60 man-hours to elicit whereas the application of the REB method took 46 man-hours.

## 5.2 Threats to Validity

Case studies come with several inherent threats to validity, particularly regarding external validity [16]. External validity concerns the extent to which the findings can be generalised beyond the setting of the study. The REB method has been applied on a small size project of ERP implementation. As such, results are limited in the extent they can be generalised but it should be noted that the case study is from a fully implemented industry example. Furthermore, the case study included a learning effect, as the project was implemented prior to the application of the REB method. Although this risk is very difficult to practically combat, we reduced the risk to the extent possible by instructing the domain experts to only answer the questions and not volunteer additional information. As such, the requirements elicited were discovered through the set of questions included in the REB method.

## 6 Related Work

Luis et al. [20] describes a method to elicit requirements from process models in three stages. First, organizational modeling is done in BPMN, then the model is validated by purpose analysis and finally functional requirement specifications are created from the refined BPMN models. It creates use-cases like specifications and suggests the elements of the model to be used in order to fill in the specification. However, the method and the requirement specification are not systematic and do not specify requirements in detail.

Use-cases and scenarios also describe the business. Maidens *et al.* [21] research aim at improving the completeness of requirements by analyzing scenarios. This process uses the existing use case model as a starting point and derives new scenarios, taking into account situations, which have not yet been considered (alternative courses). It proposes a technique to validate the completeness of models and concentrates more on the alternative paths and failure conditions. However, this approach does not hold the models as a central artifact in the elicitation process as it concentrates on the improvements and classification of requirements already gathered. However, no systematic method for eliciting requirements from a system level use-case or scenario, which was the aim of the method created in this thesis, can be found.

Meziane *et al.* [22] propose a system that generates natural language specifications from UML class diagrams. The main focus is on automatically converting models into natural language specifications. Pavlovski and Zou [23] propose a method how to formally verify informal UML Activity Diagrams. While both use UML diagrams, they concentrate on formal methods. The requirements will be in accordance with the source of elicitation and as such, inconsistent or incomplete models will result in requirement specifications of lower quality. The REB method focuses on eliciting with the aid of domain experts and therefore, produces complete requirements despite the inconsistency or incomplete models.

Some approaches include using process models in the elicitation process. For instance Demorörs et al. [24] see process models as a way to define business requirements and for creating visibility and consensus among different stakeholders.

Abeti et al. [25] suggest to use SI\*, UML and BPMN models to model organizational knowledge and use the knowledge in the RE process. These approaches recognize the usefulness of various process models but view them more for communicative purposes rather than sources of requirement elicitation.

## 7 Conclusion

The involvement of domain experts is critical when eliciting requirements. However, models used by IT are often alien and hard to understand by domain experts. On the other hand, business process models are more aligned to how domain experts see their workflows but oftentimes, such models are not complete. This paper introduces a systematic approach to elicit requirements from process models. In reality, however, such models are usually not complete. The REB method circumvents this deficiency by systematically going through different aspects of each relevant activity and thus extracting the information from the domain experts needed for a complete requirement specification. The REB method, built on mapping of components required for a complete requirement and business process models, is evaluated on a real-life case study. The results show that with less effort in terms of man-hours, the REB method results in more relevant and correct requirements being elicited as compared to the baseline set of requirements.

While other elicitation approaches value process models as communicative tools in the elicitation process, the REB method uses the process models that the domain experts understand and are comfortable with, as the foundation of systematic elicitation or requirements. As such, the elicitation process is brought much closer to the artifacts domain experts use.

Currently the REB method elicits functional requirements while non-functional requirements are not supported. Therefore, one direction for future work is to extend the method to include elicitation of non-functional requirements as well. Furthermore, the REB method would benefit from a semi-automated supporting tool that supports for instance categorization of requirements. Development of such a tool is also considered for future work.

**Acknowledgement.** This research was supported by the European Social Fund via the Doctoral Studies and Internationalisation Programme – DoRa.

## References

1. Zowghi, D., Coulin, C.: Requirements elicitation: a survey of techniques, approaches, and tools. In: Aurum, A., Wohlin, C. (eds.) *Engineering and Managing Software Requirements SE - 2*, pp. 19–46. Springer, Heidelberg (2005)
2. Coulin, C.R.: *A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation* (2007)

3. Adam, S., Riegel, N., Gross, A., Uenalan, O., Darting, S.: A conceptual foundation of requirements engineering for business information systems. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) EMMSAD 2012 and BPMDS 2012. LNBI, vol. 113, pp. 91–106. Springer, Heidelberg (2012)
4. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: ICSE 2000 Proceedings of Conference on Future of Software Engineering, vol. 1, pp. 35–46 (2000)
5. Li, J., Jeffery, R., Fung, K.H., Zhu, L., Wang, Q., Zhang, H., Xu, X.: A business process-driven approach for requirements dependency analysis. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 200–215. Springer, Heidelberg (2012)
6. Zapata, C.M., Losada, M.B., González-calderón, G.: An approach for using procedure manuals as a source for requirements elicitation. In: CLEI (2012)
7. Sutcliffe, A., Maiden, N.: The domain theory for requirements engineering. *IEEE Trans. Softw. Eng.* **24**, 174–196 (1998)
8. Sutcliffe, A., Papamargaritis, G., Zhao, L.: Comparing requirements analysis methods for developing reusable component libraries. *J. Syst. Softw.* **79**, 273–289 (2006)
9. Naish, J., Zhao, L.: Towards a generalised framework for classifying and retrieving requirements patterns. In: 2011 First International Workshop on Requirements Patterns, pp. 42–51 (2011)
10. Object Management Group Inc. Business Process Model and Notation (BPMN)
11. Wikimedia Foundation Inc. Business process
12. Dumas, M., La, R.M., Mendling, J., Reijers, H.: *Fundamentals of Business Process Management*. Springer, Heidelberg (2013)
13. Wikimedia Foundation Inc. Activity (UML)
14. Dunstan, T.: *BPMN Explained, a guide to the Business Process Modeling Notation* (2014)
15. Cockburn, A.: *Writing Effective Use Cases, The Crystal Collection for Software Professionals*. Addison-Wesley Professionals, Boston (2000)
16. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* **14**(2), 131–164 (2009)
17. Yin, R.: *Case study research: Design and methods*. Sage Publication, New York (2008)
18. Flyvbjerg, B.: Five misunderstandings about case-study research. *Qual. Inq.* **12**, 219–245 (2006)
19. Kitchenham, B., Pickard, L., Pfleeger, S.L.: Case studies for method and tool evaluation. *IEEE Softw.* **12**(4), 52–62 (1995)
20. de la Vara, J.L., Sánchez, J., Pastor, Ó.: Business process modelling and purpose analysis for requirements analysis of information systems. In: Bellahsene, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 213–227. Springer, Heidelberg (2008)
21. Maiden, N.A.M., Minocha, S., Manning, K., Ryan, M.: CREWS-SAVRE: Systematic Scenario Generation and Use 1 2: The CREWS-SAVRE Software Tool, pp. 1–9
22. Meziane, F., Athanasakis, N., Ananiadou, S.: Generating natural language specifications from UML class diagrams. *Requir. Eng.* **13**, 1–18 (2007)
23. Pavlovski, C.J., Zou, J.: Non-functional requirements in business process modeling. In: APCCM, vol. 79, pp. 103–112 (2008)
24. Demirörs, O., Gencel, Ç., Tarhan, A.: Utilizing Business Process Models for Requirements Elicitation, pp. 1–4 (2003)
25. Abeti, L., Ciancarini, P., Moretti, R.: Business process modeling for organizational knowledge management. In: Degano, P., De Nicola, R., Meseguer, J. (eds.) *Concurrency, Graphs and Models*. LNCS, vol. 5065, pp. 301–311. Springer, Heidelberg (2008)