Requirements Elicitation with Business Process Modeling

LISE B. HVATUM

Requirements Elicitation is the area of Requirements Engineering that deals with identifying system requirements. The paper documents a pattern showing how to elicit functional requirements for a software product in the form of User Stories through modeling the operational processes to be performed by users of the system. Using business process modeling to identify functional requirements to obtain a complete and consistent set of requirements for the scope of the model. The pattern is placed into the overall context of requirements elicitation techniques.

Categories and Subject Descriptors: **D.2.1 [Software Engineering]**: Requirements/Specifications—*Elicitation Methods.*

General Terms: Management

Additional Key Words and Phrases: requirements engineering, requirements elicitation, requirements analysis, business process modeling

ACM Reference Format:

Hvatum, L. 2014. Requirements Elicitation using Business Process Modeling. 21st Conference on Pattern Languages of Programming (PLoP), PLoP 2014, September 14-17 2014, 9 pages.

1. INTRODUCTION

"The hardest single part of building a software system is deciding precisely what to build.
No other part of the conceptual work is as difficult as establishing the detailed technical requirements,
including all the interfaces to people, to machines, and to other software systems.
No other part of the work so cripples the resulting system if done wrong.
No other part is more difficult to rectify later."
Frederick Brooks in "No Silver Bullet: Essence and Accidents of Software Engineering"

The system requirements for a software product are the basis for core development activities – they drive architecture, User Interface design, code, testing, and documentation. The best developers with the best process and tools available will still fail in delivering a good product unless the requirements and their implications are well understood at the time when they are consumed by the development team.

Requirements elicitation is the part of the Requirements Engineering that deals with identifying, clarifying and capturing the system functionality. Literature on requirements elicitation [Hossenlopp et.al. 2008, Masters 2010] present variations and/or subsets of the following techniques to extract requirements:

Interviews – normally done with stakeholders and users starting with a defined set of questions but with the possibility to expand and elaborate the discussion to further explore the needs.

Questionnaires – have a defined set of questions and typically reaches out to a larger group of individuals than the interviews.

Workshops – bring together a small number of users and stakeholders to brainstorm on functionality and ideas for the product.

Analysis of existing systems – if the new product is replacing or will compete with existing systems a lot can be learned from studying the existing system and its documentation.

Observation – of users performing tasks either manually or with an existing system.

Prototypes and pilots – limited implementations can give users early experience to validate requirements as well as providing additions and changes to already defined requirements.

This paper explains an additional approach to eliciting functional requirements for a software product from the perspective of the users through modeling the operational (user) workflows, more specifically by using a formal business modeling (BPM) technique.

The target audience of this paper is the Business Analyst role and other roles involved in developing the functional requirements for a software product from high level business needs. Readers who are not familiar with Business Process Modeling (BPM) and its formal notation (BPMN 2.0), or with the formal definition of User Stories as used in Agile processes, should read the appendix before proceeding.

Over the last five years the author was involved in two major development projects. The first was an IT business system to support the engineering operation of a major company (i.e. several thousand users) as this organization was undertaking a major process change. The activity to model the new operational workflows was so large that BPM experts were brought in to do the modeling. The second project was an automation project where current human workflows would be replaced by automated systems. Here the workflow modeling was done by operational experts. Common to both projects was the usefulness of the many models to analyze and extract the functional requirements from the workflow drawings.

A literature search revealed several papers on elicitation of requirements using models [Aslan 2002, Decreus et.al. 2009, Demirörs et.al. 2003, Dragicevic et.al. 2011, Monsalve et.al. 2011, Monsalve et.al. 2010, Velarde 2013]. Unfortunately, these are mostly research papers with a rather narrow focus. So while extracting requirements from models (and more specifically from business process models) is a method known from the literature, the author did not find what she was hoping for – a practical description of how to elicit requirements from models that easily could be applied by software teams. This paper is an attempt to fill that void.

It is important to point out that one would normally apply a combination of requirements elicitation techniques to reach a good set of requirements, and that it is an iterative process where techniques are repeated throughout development to elaborate and refine initial ideas (agile). One possible flow is shown in Figure 1 with an indication of knowledge entities that would gradually be refined (for example user tasks). This is of course a simplistic view, as the team of modelers and analysts would typically repeat techniques at any time they need more information, for example doing additional interviews after the first workflow models are done to gain additional insights or reach out to a new of larger set of users. But it does give an idea of where the workflow modeling fits in the overall undertaking of requirements gathering.

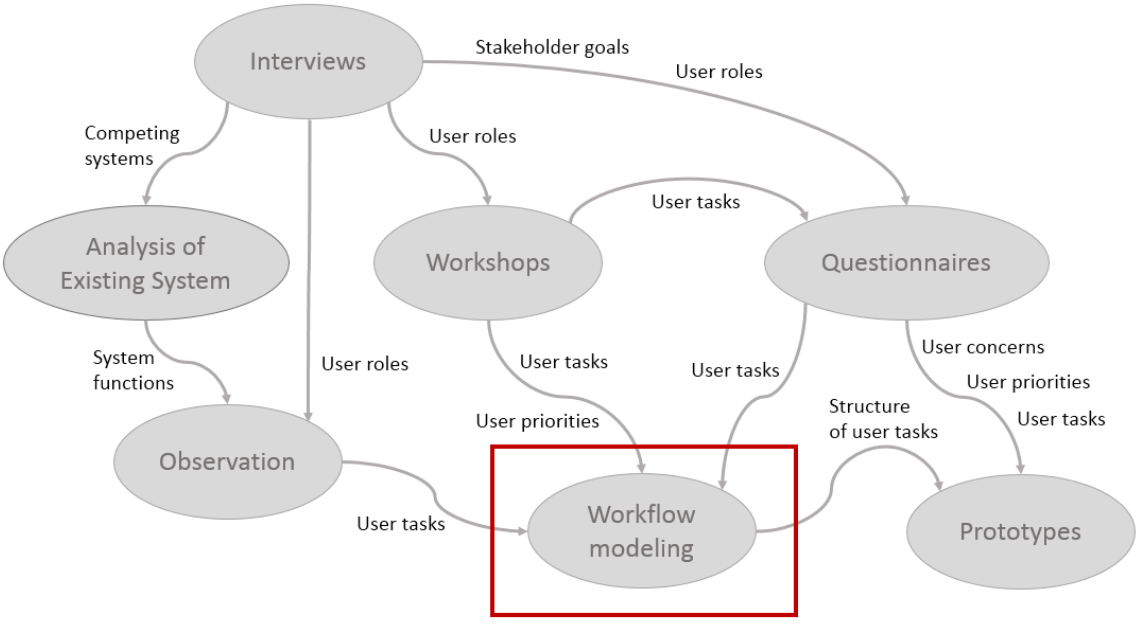
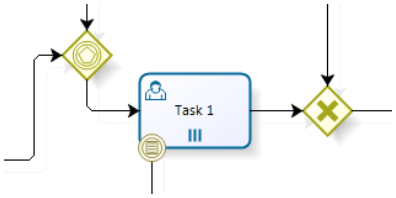


Figure 1: Flow of requirements elicitation techniques



User Stories from Workflow Models

"Developers have long recognized that understanding of the business processes is the key to eliciting the needs of their users." *-- Dragicevic, Celar, Novak*

Business Process Modeling can be used to elicit detailed functional requirements.

***

After the business managers have provided a product vision and some high level business needs the business analysts and software development team are usually left with most of the work to hash out detailed requirements and software specifications. They will do interviews, workshops and other activities that provide a thorough understanding of the business needs. These need to be translated to detailed functional requirements to populate the development backlog each time the team starts development of a new product increment. The team is familiar with User Stories and this is their preferred way to capture functional requirements.

How can a software development team get a complete and consistent set of functional requirements for a product increment from the product vision and high level requirements?

Brainstorming sessions are often done to capture User Stories. They can generate a lot of good material that then need to be sorted and structured before a final prioritized list can be made. But by nature these brainstorming sessions tend to lack structure and do not ensure that all the needed functionality is properly covered.

User interviews are another common technique. Again the structure of the problem may not be evident from the user feedback. Each user will cover their own role but may not be fully aware of dependencies to other users and their activities.

Access to the users is a common problem for development teams, and the time allotted by the users to communicate with the team must be utilized as well as possible.

The needs of more peripheral user roles like system administrators are sometimes included late and functionality for these users may therefore be somewhat sketchy and/or become a challenge with the current architecture that was defined without incorporating the needs of these roles.

In some cases the system to be developed is to support new activities for the users, or a change in how activities are being performed. Current users may not know or understand the new vision, while the business managers who own the vision may not know the details of the operation of the users.

Therefore:
Create business process models of the operational workflows derived from the business requirements for the product increment, then analyze the models to identify the functional requirements for each role.

As shown in Figure 2, the starting point for the modeling is to get a good definition of the user roles that will directly interact with the system. The next step is to create a list for each user role defining the core activities the role is involved in, and what categories of information the role needs to perform the activities. By reviewing the collated set of user activities across the roles it will be possible to identify a set of business processes to model.

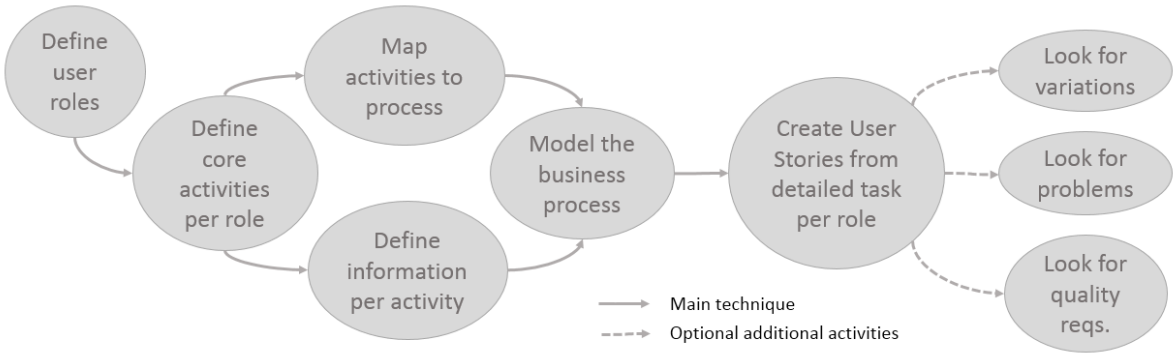


Figure 2: Elicitation process

To show how functional requirement (here in the form of User Stories) are derived from the process model consider the below example. Figure 3 shows a process workflow for hiring a new employee. There are three roles defined: the Candidate, the Recruiter, and the Interviewer, each with a separate swim lane. The tasks for each user are shown in the swim lane for that specific role.

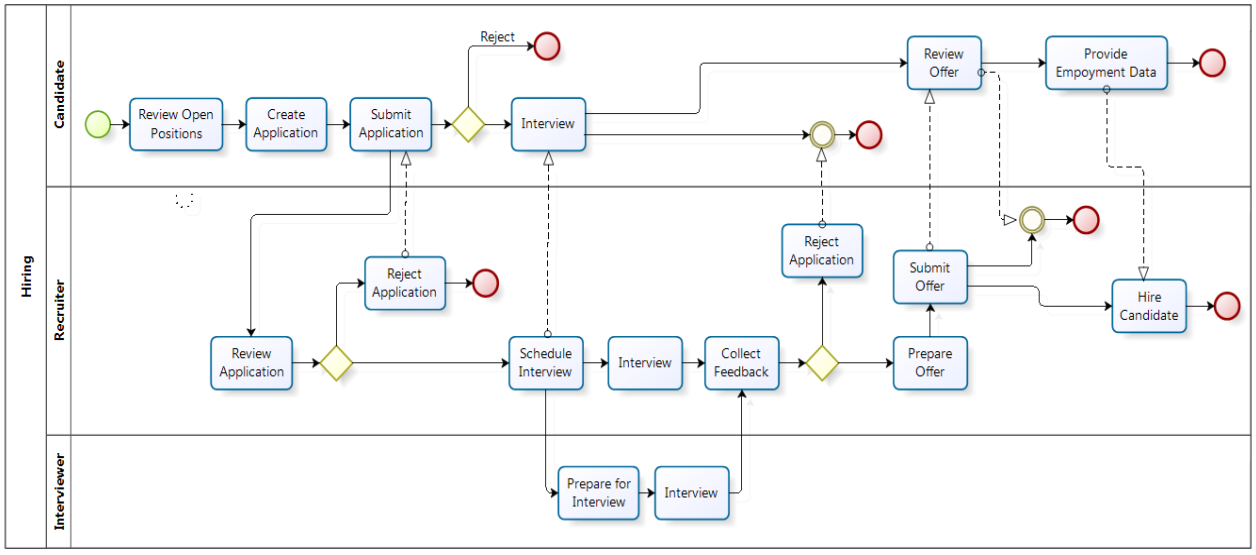


Figure 3: Workflow for a hiring process using BPMN

By analyzing the individual swim lanes in the figure, User Stories are easy to identify. Here is a (non-exhaustive) list of User Stories for the Candidate role:

- As a Candidate I can see open positions
- As a Candidate I can submit an application

- As a Candidate I get notified of a rejected application
- As a Candidate I can receive an invitation for an interview
- As a Candidate I can receive an offer for a position
- As a Candidate I can accept the offer for a position
- As a Candidate I can reject the offer for a position
- As a Candidate I can provide employment information

Next, a deeper analysis of the workflow preferably in discussion with users/customers can lead to additional User Stories. What if the Candidate does not see any open positions, can he/she still apply? What if the candidate wants to cancel the application after it is submitted? Or add additional information to it? The workflow model helps to explore variations and special cases even if these are not explicitly modeled. Most modeling tools will support adding comments to the process elements to capture discussions and questions from the development team.

Creating User Stories from a BPM workflow provides a structured approach for requirements elicitation that increases the chances to get a high quality set of requirements. Carl Wiegers lists the following criteria for excellent requirements specifications (sets of requirements) [Wiegers 2009]:

- Complete – no missing requirements
  By covering the complete workflow in the analysis the resulting set of requirements will represent all the user actions needed to execute the workflow. Normally they need to be developed as a comprehensive set to support a transaction, or a system slice, to bring business value.

- Consistent – no conflict with other requirements
  The User Stories will tie together in a natural flow given by the sequence of user tasks.

- Modifiable – ability to revise and maintain a history of changes
  Any change in a User Story can be checked against tasks in the same workflow and any change can then be done consistently on the versioned set of requirements that belong to the same workflow.

- Traceable – linked to origin
  Each User Story has a clear origin from a workflow task.

If the requirements contents are structured with core high-level requirements being represented by workflows the outcome of the analysis will not only be the detailed requirements but also the overall requirements structure. User stories can be sorted both on the high-level requirements and by the role involved.

By understanding the operational workflow the User Stories enable it is possible to discuss business priorities with the customer on a higher level than for each separate functional requirement. Because the modeling is done using terminology from the problem domain (business), the dialogue with the sponsors and user representatives will be easier. Broken down to the right level of granularity the workflows will bring good insights for developers in understanding the product.

The main challenge with this method is to decide what workflows to model, and at what level of granularity. This is similar to deciding for Use Cases – if they are too high level you do not gain enough insight, and if they are too detailed you will get a lot more work without the pay-off. To be able to gather detailed user requirements the workflows need to be fairly low level. Until a team has some experience they will need to experiment with the workflows to find the best level of detail for their type of application. The goal is to model to a level of detail that is enough to fully understand the problem but no deeper. Each workflow should represent a fairly significant amount of work, which is reflecting a major transaction that fulfils a business level feature of the system. For example, one workflow model would encompass all actions in buying a book online from searching to completing the checkout.

Another challenge is how to handle variations in the workflow. For some types of operation there may be a multitude of action sequences that can lead to the completion of the transaction, and many more when considering possible problem situations. To benefit from the workflow model it is not necessary to model all possible variations. These can come later through use cases, scenarios and other requirements artifacts. It is better to let the workflows represent the major happy scenario for a transaction but with clear indications of were the happy scenario could break. One may include some lower level workflows where these bring additional clarity to the team.

Developing operational workflows is time consuming and demand access to operational experts. If the workflows are complex it may be best to hire resources with expertise in business process modeling. On the tools side there are free tools that are good to create simple models with very little training. There are also highly sophisticated tools for BPM that are costly but may be worth the investment if the process models are to be kept for a long time, and if they are layered and create a system of systems wholeness.

Unless the organization is doing process modeling for other reasons than the software requirements it should be discussed for how long the process model should be kept. It may be best to use it only as an initial tool to build product understanding and gather detailed requirements. One will find it hard to update the process models as the software evolves and gets modified based on user feedback. The models must be kept free of the technical and software solutions – focus on the user tasks and operations without referring to how this may be implemented in the current (to-be-replaced) or future system.

The need for input from the users is not reduced, but the method can help structure the discussions and make the best use of the time that the users have available. Workflows that are validated by users are usually easy to understand by the business owners. It is recommended to have them approve the workflows as part of approving the business requirements.

This technique is used to find the functional requirements of a product as experienced by a user. It will not directly provide non-functional requirements although the workflows can be followed to discuss performance of various operations, the need for security, etc. The workflow understanding may influence the User Interface design to help improve the usability.

2. COMMENTS

This paper is a result of the authors' experience using workflow modeling to elicit requirements. The literature study done when working on this paper gave interesting leads to other related work, and helped seeing how this pattern can be placed within a larger context. As correctly pointed out by the shepherd, the pattern in this paper has a wide scope and could possibly be broken up into smaller patterns. The author plans to continue exploring requirement elicitation practices and as part of this future work will rework the current pattern into a set of lower level patterns.

3. ACKNOWLEDGEMENTS

The author is thankful to my previous manager Tom Provost who introduced me to the domain of business process modeling, and to my Business Analyst Kenya Dixon who is a great discussion partner on techniques and tools for requirements. A big thanks goes to my shepherd Hans Wegener who was willing to pick up a very sketchy paper and has taken me through a rightfully painful but so very helpful process – I appreciate all your time and your questions and comments that will continue to guide this work! The writer's workshop at PLoP 2014 provided additional comments that have helped clarifying and improving this paper, and that will be very helpful for the future work as mentioned above.

4. APPENDIX

To provide a contextual background for the pattern it is necessary to introduce the Business Process Modeling (BPM) and its formal notation (BPMN 2.0), and to point out some concepts from software requirements engineering as they apply in this paper.

4.1 Business Process Modeling

Business Process Modeling is a well-established technique to help businesses understand their internal operational workflows [Wikipedia 2014a]. The formal knowledge domain of BPM came to life in the 1990s but builds on a long history of models and theory about work processes (flow diagrams, Gantt charts). Although the core purpose of BPM is to analyze and document, and even design, business workflows for an operation, the use of BPM to elicit requirements for software development is now a referenced method [Aslan 2002, Decreus et.al. 2009, Demirörs et.al. 2003, Dragicevic et.al. 2011, Monsalve et.al. 2011, Monsalve et.al. 2010, Velarde 2013]. It was not a surprising development as companies wanted IT business systems to support and possibly automate their desired workflows.

The current BPMN 2.0 provides a standard modeling notation that facilitates communication around business processes internally as well as with other involved parties like IT solution providers, supply chain, and vendors. Below is a short description of BPMN to help the reader understand this paper. More complete information about BPMN can be found on Wikipedia [Wikipedia 2014b] or on the Object Management Group page about BPMN [OMG 2014].

A very simple process model using BPMN is illustrated in Figure 4. The process is represented by a *Pool* with a *Swim Lane* for each role that participate in the process. The main element is a *Task* placed in the swim lane of the role that performs this task. A line with arrows connecting tasks called *Sequence Flow* represent the flow of the operation. A process must have at least one *Start Event* and one *End Event*, but multiple start and end events may be present. Tasks may have associated *Data Objects* linked with an *Association* dotted line and arrow.

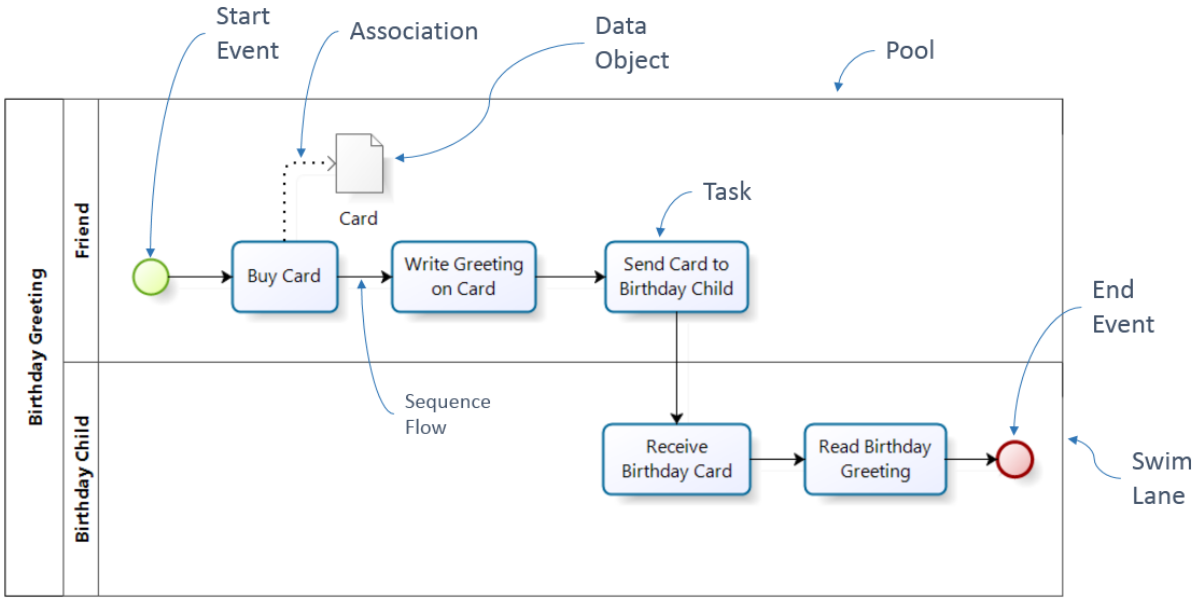
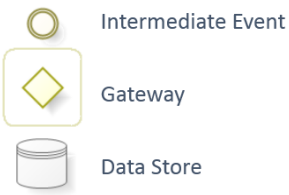


Figure 4: Process example

There are additional notational entities in BPMN of which the following should be mentioned here to complete the BPMN introduction:



An *Intermediate Event* has an impact on the process flow

A *Gateway* represents a decision where the flow can take two or more alternate paths

A *Data Store* is another type of object that can be associated with a task

The BPMN notation is easy to pick up and apply. There are several tools available from basic free products like **Bizagi** [Bizagi 2014] (used for all illustrations in this paper) to sophisticated modeling applications like OpenText **ProVision** [ProVision 2014].

4.2 User Stories

A great source to learn about requirements engineering is the book Software Requirements by Karl Wiegers [Wiegers 2009]. Wiegers outlines a set of activities that are part of the elicitation, among which are “Identify user classes”, “Identify use cases”, “Hold facilitated elicitation workshops”, and “Observe users performing”.

User Stories [Cohn 2004, Ambler 2014] are today a common technique used to capture functional requirements, which are requirements that “describe the observable behaviors the system will exhibit under certain conditions and the actions the system will let users take” [Wiegers 2009].

The User Story follows a specific format:

As a (role) I can (action) so that (purpose).

There are some variations in form as shown in the below examples. The format is designed to fit on an index card. Typically the User Story is on the front of the card, and the back of the card is then used to capture acceptance criteria, business priority, and estimated effort. Most software development tools support the user story format electronically (for example in TFS from Microsoft when using the agile template).

User story examples:

- As a friend I can buy a birthday card
- As a friend I can send a birthday card so that I can greet the birthday child on her birthday
- A birthday child receives greetings cards

REFERENCES

AMBLER, S. 2014. http://www.agilemodeling.com/artifacts/userStory.htm

ASLAN, E. 2002. *A COTS-Software Requirements Elicitation Method from Business Process Models (Thesis)*. Middle East Technical University (METU)

BIZAGI 2014. http://www.bizagi.com/

COHN, M. 2004. *User Stories Applied*. Addison-Wesley (ISBN 0-321-20568-5)

DECREUS, K., EL KHARBILI, M., POELS, G. AND PULVERMUELLER, E. 2009. Bridging Requirements Engineering and Business Process Management. In *Proceedings of Software Engineering Workshopband, Fachtagung des GI-Fachbereichs Softwaretechnik*.

DEMIRÖRS, O., GENCEL, Ç. AND TARHAN, A. 2003. Utilizing Business Process Models for Requirements Elicitation. In *Proceedings of the 29th EUROMICRO Conference, IEEE.*

DRAGICEVIC, S., CELAR, S. AND NOVAK, L. 2011. Roadmap for Requirements Engineering Process Improvement using BPM and UML. In *Advances in Production Engineering & Management 6*, 221-231.

HOSSENLOPP, R. AND HASS, K. 2008. *Unearthing Business Requirements: Elicitation Tools and Techniques*. In Management Concepts (ISBN 978-1-56726-210-0).

MASTERS, M. 2010. *An Overview of Requirements Elicitation*. http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/1427/An-Overview-of-Requirements-Elicitation.aspx

MONSALVE, C., APRIL, A. AND ABRAN, A. 2011. Requirements Elicitation Using BPM Notations: Focusing on the Strategic Level Representation. In *Recent Researched in Applied Computer and Applied Computational Science.*

MONSALVE, C., APRIL, A. AND ABRAN, A. 2010. Representing Unique Stakeholder Perspectives in BPM Notations. In *Eighth ACIS International Conference on Software Engineering Research, Management and Applications*, 42-49.

OMG 2014. *Business Process Modeling Notification*. http://www.bpmn.org/

PROVISION 2014. http://www.opentext.com/what-we-do/products/business-process-management/process-suite-add-ons/opentext-provision-for-enterprise-architecture

VELARDE, L. 2013. *Using BPMN to gather Requirements for Enterprise Software Implementations*, http://www.leverforce.com/index.php/insights/cbap-blog/153-as-is-and-to-be-bpm-for-erp

WIEGERS, C. 2009. *Software Requirements* 2nd Edition. Microsoft Press (ISBN: 0-7356-3708-3).

WIKIPEDIA 2014a. Business Process Modeling. http://en.wikipedia.org/wiki/Business_process_modeling

WIKIPEDIA 2014b. Business Process Model and Notation. http://en.wikipedia.org/wiki/Business_Process_Model_and_Notation