Business Process Model Driven Automatic Software Requirements Generation

Salam Turkman and Adel Taweel(✉)

Computer Science, Birzeit University, Birzeit, Palestine
salma.turk@gmail.com, ataweel@birzeit.edu

Abstract. Requirement engineering is a critical stage in software engineering, it enables requirement engineers extract correct system needs, both functional and non-functional constraints from stakeholders. The majority of the errors found in software functionality are directly linked to the mistakes made during the requirement elicitation phases. Therefore, several approaches have been proposed to enhance existing requirements engineering techniques to both reduce such mistakes and to speed up the requirements engineering process. One type of promising approaches is based on utilizing business process modelling to take benefit from business process models to derive requirements. This paper argues that it is possible to generate requirements from business process models. It proposes an approach to derive system requirements; it employs business process models and then transforms them into requirement models. Evaluation shows the proposed approach was able to generate additional valid use case model features compared to other competing approaches.

Keywords: Requirement engineering · Business process modelling · Use case model

1 Introduction

Requirement engineering is a critical stage in software development [11]. Employing effective requirements engineering techniques and methods are essential to the success of software development projects, not only for achieving them on time and within budget but also for delivering the desired business value [12]. Research has shown that many large projects fail because of inadequate requirements, showing that errors made in the requirements engineering stage "are among the most difficult to detect and the most expensive to correct" [5]. These errors are caused by problems that mostly appear in the requirement elicitation phase [15, 16].

Many approaches have been proposed to enhance the existing requirements engineering techniques; some of these approaches recognized that "understanding a business process is the key to identify the user needs of the software that supports it" [13, 19, 20, 22]. Many organizations have their existing business process models in the form of working instructions and often include enough valid details for specifying software systems, which thus may provide a basis for understanding and modelling software requirements. In recent years, requirements of business applications have changed from command-based applications to workflow-based applications, "at least half of industrial software development is connected to business application development" [21]. Thus business process models may provide a promising approach to help produce accurate requirement specifications [10, 14, 24].

Several approaches have proposed methods to automatically generate requirements specifications from business models but these approaches fail to achieve transformation without significant manual intervention or correction [1–4, 13]. This paper argues that not only business process models can be used to derive more accurate software requirements but also these requirements can in majority be generated automatically. It proposes a new business model-driven approach for deriving UML-based requirement specifications. The proposed approach employs a set of systematic steps that start by improving the existing business process model to result into a well-defined business model, which includes the effect of the prospective information system should have on the business processes (named "To-Be" model). It then automatically transforms the "To-Be" business process model to a Use Case diagram. The proposed automatic generation of valid requirement specifications overcomes the often-used tedious and time-consuming manual process. Initial evaluation shows accurate results when compared with other manual approaches; it also shows the generation efficiency is directly proportional to the level of richness of the input business process model.

2 Related Work

Several researchers have proposed approaches to derive use cases diagram from business process models [1–4, 13, 22]. [1] Proposed an algorithm to automatically transform business model in to functional requirements in terms of use case model. The main objective for this approach is to draw up functional requirement more quickly from existing business model instead of building a use case diagram depending on interviews. The algorithm works by first creating meta-models for both the use case diagram and business process model then compares definitions in the two meta-models, to find "which concepts or relations in business process meta-model map to which concepts or relations in the use case meta-model". This approach was evaluated by comparing their results with use cases constructed by performing interviews, although promising however the total error percentage in the generated use case diagram was relatively high at 40%. Another similar study [2] proposed a method to explore associations between the use case model and the business model, but they used "Role Activity Diagrams" or RADs to model business processes. However, they faced several issues in deriving use cases from process models, including the notion of an actor, which is not clear enough in RADs. They found no simple mapping of Roles, in process models, on to Actors in use case diagrams [2]. Their results show, however, the transformation cannot achieve a well-formed use case model because the RAD notation is incompatible with UML-they found that UML actor is often not clearly defined in a RAD.

This work is further developed by [3], however they proposed a method for deriving system models based on business process models. They suggested that the correspondence between the central notion of 'automated activity' in an improved RAD model and that of "action or function" in the use case diagram facilitates the derivation of system models based on business process models. Their method consists of four steps: 1-develop a business process model using RAD model; 2-identify automated activities; 3-link each business objective with automated activities, and 4-develop use case model based on objectives and automated activities. However, this approach have shown several limitations, including lack of process visibility, and focus on use cases only with no notation of associations between them.

Another approach [13] proposed a manual transformation to obtain use case model based on business process models. This approach is the first approach that attempted to generate use case description from business process models. However, it focused on generating the use case description more than the use case diagram noting the value that may be obtained from the description. The use case diagram was generated manually depending on a set of rules, the resultant use case diagram however was not detailed enough. Generated use case diagrams, by this approach, did not cover the association between use cases such as extend, include, invoke and precede. The use case descriptions were specified from a set of predefined natural language sentences mapped manually from BPMN model elements. [4] Proposed a similar manual approach for deriving system requirements from business process models. This approach integrates requirements engineering and business process engineering. It defined BORE: a Business-Oriented approach to Requirements Elicitation. The authors argue that BORE is especially effective when system requirements are not fully knowable up front and must be discovered. [22] studied the use of DEMO (dynamic essential modelling of organisations) for deriving use cases from business models, and investigated most suitable methods or ways for identifying suitable use cases. The suggested methods can be used in combination of the automatic generation approaches to enhance use case derivation.

3 Proposed Approach: BMSpec

Generating valid and useable requirements from business models requires having valid business models. To achieve, BMSepc developed two original methods to enable consistent derivation of use cases from business models. It developed a structured and systematic "to-be" business model preparation and a set of heuristic rules that define the derivation and transformation of uses cases. The structured and systematic method ensures validity of the business model through the consistency of constraints to the correctness of its representation. As mentioned above, BMSpec employed requirement engineering and business model engineering integration from [4] for the "As-Is" business model part. BMSpec consists of two systematic steps as shown in Fig. 1: first, preparation of a well-defined business model. This step requires manual processing and its required effort depends on the status of the existing business process model. Second, automated transformation of business process model to a UML use case model. This step uses a BMSpec developed algorithm to map XML objects in both models. The developed algorithm employs a defined set of heuristic rules that define transformation of objects and relations from business process model to a use case diagram. These steps are described in further details in the following subsections.

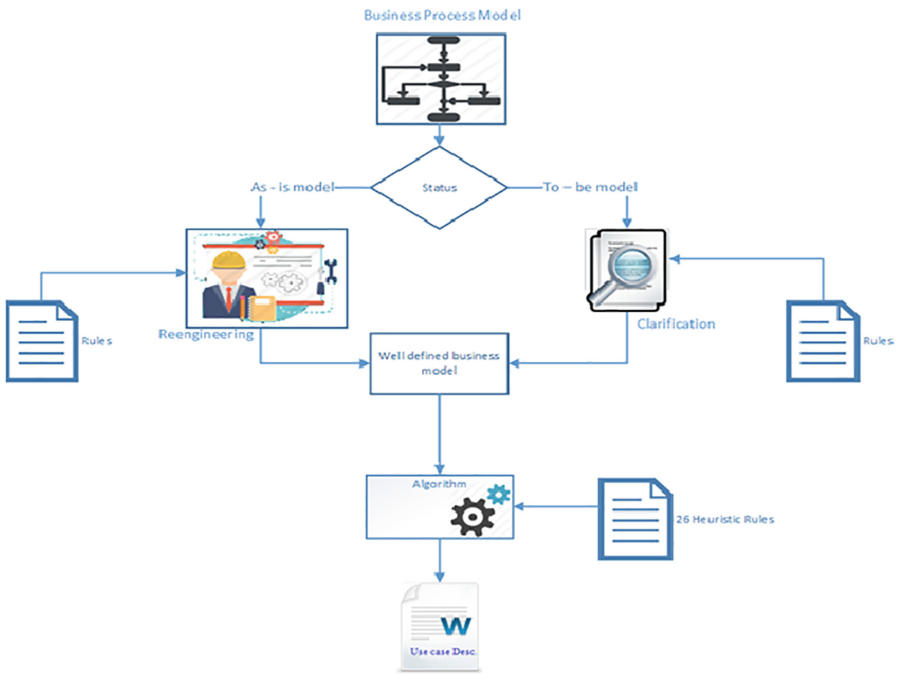


Fig. 1. BMSpec - overall structured approach

3.1 Preparation of a Well-Defined Business Process Model

A manual transformation of the existing business process model is undertaken to ensure model validity and consistency. The level and required effort of transformation depends on the status of the business process model. In cases where the business model is just a manual “As_Is” business process model and does not cover the user interaction with the system, the model needs reengineering in order to build a (To_Be) business process model. The transformation, in this case, includes analysing the purpose of the system and determining the effect of the information system should have on the business processes (To-Be). The key output of this step is to determine the automated tasks that represent user interactions with the system. However, simply automating processes for the sole purpose of automation often does not result into significant improvements [6]. Thus, in business process reengineering, it is recommended that instead of blindly automating manual processes, processes need to be reengineered while taking advantages of the possibilities for automation [7]. Business process reengineering may require “reshaping the way business is done” [8]. This further requires taking an integrated business look at both process and information flows, including looking at how processes use information and how people interact with systems [8].

In order to reengineer the business process model effectively, this reengineering step should focus on both system perspective, i.e. asking “what will make up a well-defined use case?”; and business process perspective, i.e. asking “what is needed from the information system?” [9].

In BMSpec, in this step, the aim is to identify the tasks that represent interactions with the software system. Because these tasks often represent the key important functions, or use cases, that must be included in the use case model. Thus, each well-defined use case must specify a functionality, which an actor wants to achieve by using the system. Assuming the (As-Is) business model is represented using the BPMN, to guide this step, BMSpec defines a set of rules to achieve the business process (To-Be) model (discussion of these rules is outside the scope of this paper). For business process models that are already designed with a clear software system purpose, and a clear effect of the information system on the business processes (To-Be), but have some BPMN notations that are not clearly defined, e.g. specifying task type, or declaring condition for gateways, events name and type, the business process model's notation representation is required to be modified. To conform to BPMN notations, BMSpec defines another set of rules to guide this modification (outside the scope of this paper).

3.2 Use Case Diagram Generation: Heuristic Rules

Once a business process To-Be model is created, BMSpec defines a set of heuristics that are used to identify, transform and generate actors, use cases, and their associations into a use case diagram. These heuristic rules have been developed based on studying the BPMN language and its notations, and their semantic use and meaning, and analysing more than 70 real-time business models to arrive at consistent transformation and interruptions of BPMN notations and their combinations. These heuristic rules have developed into computational algorithms to automate the business process To-Be models transformation and use-case generation into a UML use-case diagram. Description and listing of defined heuristic rules are outside the scope of the paper.

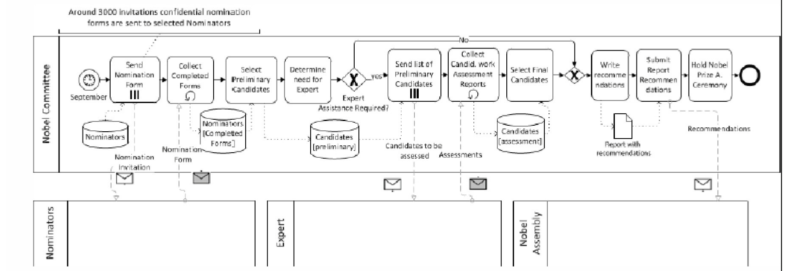
3.3 Automatic Use Case Generation: Algorithm

To automate the process of use case generation from the modified business process models, an algorithm has been developed that automatically reads the business process model, provided as BMPN notation, generates use case diagram as output. It performs three main steps:

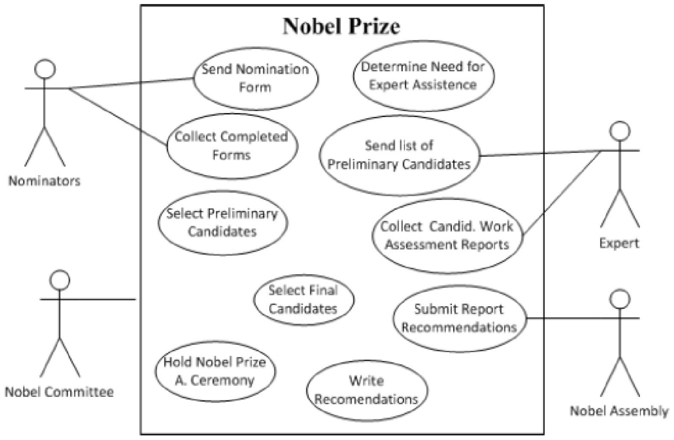
1. identifies the business process models activities and nodes
2. identifies processes, workflows, gateways and conditions to ensure consistency and connections between its nodes.
3. applies its defined heuristic rules, on each of the identified activities and maps them to respective use cases. Connectors and actors are also consequently generated.

Table 1.

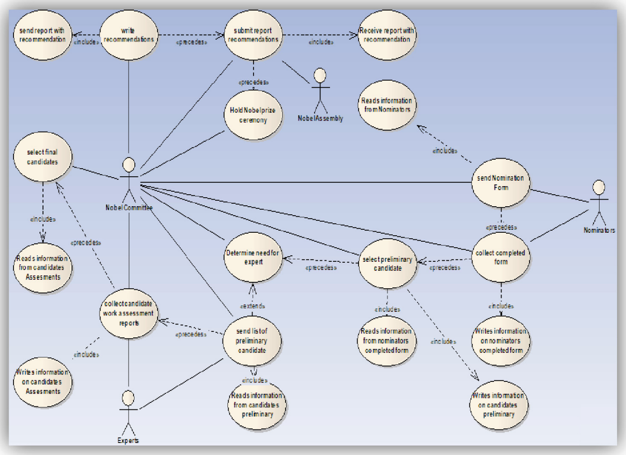
Case study name	Brief description
Nobel Prize example	It is a real case in which a paper work [13] used manual transformation from business process model for Nobel prize to a use case model. In the manual transformation authors used a set of rules


Around 3000 invitations confidential nomination forms are sent to selected Nominators
Nobel Committee
September
Send Nomination Form
Collect Completed Forms
Select Preliminary Candidates
Determine need for Expert
Expert Assistance Required?
yes
No
Send list of Preliminary Candidates
Collect Candid. work Assessment Reports
Select Final Candidates
Write recommendations
Submit Report Recommendations
Hold Nobel Prize A. Ceremony
Nominators
Nominators [Completed Forms]
Candidates [preliminary]
Candidates [assessment]
Nomination Invitation
Nomination Form
Candidates to be assessed
Assessments
Report with recommendations
Recommendations
Nominators
Expert
Nobel Assembly

Business process model.


Nobel Prize
Send Nomination Form
Determine Need for Expert Assistence
Collect Completed Forms
Send list of Preliminary Candidates
Select Preliminary Candidates
Collect Candid. Work Assessment Reports
Select Final Candidates
Submit Report Recommendations
Hold Nobel Prize A. Ceremony
Write Recomendations
Nominators
Expert
Nobel Committee
Nobel Assembly

Manually developed use case diagram.


send report with recommendation
write recommendations
submit report recommendations
Receive report with recommendation
«include»
«precedes»
Hold Nobel prize ceremony
Nobel Assembly
Reads information from Nominators
select final candidates
Nobel Committee
send Nomination Form
Nominators
Reads information from candidates Assesments
Determine need for expert
select preliminary candidate
collect completed form
«extend»
collect candidate work assessment reports
send list of preliminary candidate
Reads information from nominators completed form
Writes information on nominators completed form
Writes information on candidates Assesments
Reads information from candidates preliminary
Writes information on candidates preliminary
Experts

BMSpec generated use case diagram

Fig. 2. Noble prize case study

4 Results

The traditional gold standard testing or model evaluation is used [18]. This is conducted by evaluating the output of traditional requirement elicitation techniques, in which software engineers are employed to build use case diagrams manually, against BMSpec generated use case diagrams, for the same scenarios. Although this is an expensive procedure, yet to ensure validity, the evaluation was done on several different case studies.

As an example of the conducted evaluation and results, Table 1 lists one example of an evaluated case study. Figure 2 shows an example of this case study [13], including its business process model, manually developed use case diagram and its BMSpec generated use case diagram. The case study used traditional requirement engineering elicitation techniques to develop their requirement specification or use case diagram, and manual transformation from business process model to requirement specifications. As shown in Fig. 2, BMSpec was able to identify and extra features, which are not supported in other competing approaches [1–4, 13], such as association between use cases (precede, invoke, include, extend). Detailed evaluation is outside the scope of the paper.

5 Conclusion

The paper argues that business models can be usefully used to generate accurate requirements and software specifications. To achieve, the paper developed a systematic approach (BMSpec) that takes a number of systematic steps using standardized BPMN notations. Evaluation and illustrative results of the proposed BMSpec approach are described and compared against manual traditional requirement engineering techniques. It shows the promise of the approach and higher generation efficiency, which was found to be directly proportional to the level of richness of the input business process model.

While the proposed approach improves the efficiency of the automatic generation of UML-Based use case diagram, it does not however cover or replace the entire requirements engineering stage, nor aims to generate comprehensive requirement specifications. It provides, however, an important step forward to semi-automate the elicitation process through the extraction of as many as possible of requirements from underlying business process models, thus potentially significantly saving development time, reducing requirement misunderstanding errors and improving correct requirements representation using software industry de facto UML.

References

1. Dijkman, R.M., Joosten, S., Ordina, F.: An algorithm to derive use case diagrams from business process models. In: Proceedings of the 6th International Conference on Software Engineering and Applications (SEA), Anaheim, US (2002)
2. Odeh, M., Richard, K.: Bridging the gap between business models and system models. Inf. Softw. Technol. **45**(15), 1053–1060 (2003)
3. Aburub, F.: Activity-based approach to derive system models from business process models. In: 2012 International Conference on Information Society (i-Society). IEEE (2012)
4. Przybylek, A.: A business-oriented approach to requirements elicitation. In: 2014 International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE). IEEE (2014)
5. Boehm, B.W.: Software Engineering Economics, vol. 197. Prentice-Hall, Englewood Cliffs (1981)
6. Weerakkody, V., Currie, W.: (2003) Integrating Business Process Reengineering with Information Systems Development: Issues & Implications. In: van der Aalst, W.M.P., Weske, M. (eds.) Business Process Management. BPM 2003. Lecture Notes in Computer Science, vol. 2678, pp. 302–320. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-44895-0_21
7. Hammer, M.: Reengineering work: don't automate, obliterate. Harvard Bus. Rev. **68**(4), 104–112 (1990)
8. Kaplan, R.B., Murdock, L.: Rethinking The Corporation: Core Process Redesign, The Mckinsey Quarterly, 2 November 1991
9. Eriksson, E., Magnus, P.: Business Modeling with UML. Business Patterns at Work. Wiley, New York (2000)
10. Dijkman, R., Hofstetter, J., Koehler, J. (eds.): BPMN 2011. LNBIP, vol. 95. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25160-3
11. Pressman, R.S.: Software Engineering: A Practitioner's Approach. Palgrave Macmillan, New York (2005)
12. Bloch, M., Sven, B., Jürgen, L.: Delivering large-scale IT projects on time, on budget, and on value. McKinsey Q. (2012)
13. Cruz, E.F., Machado, R.J., Santos, M.Y.: From business process models to use case models: a systematic approach. In: Aveiro, D., Tribolet, J., Gouveia, D. (eds.) EEWC 2014. LNBIP, vol. 174, pp. 167–181. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06505-2_12
14. Indulska, M., Recker, J., Rosemann, M., Green, P.: Business process modeling: current issues and future challenges. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 501–514. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02144-2_39
15. Rajagopal, P., Lee, R., Ahlswede, T., Chiang, C.C., Karolak, D.: A new approach for software requirements elicitation. In: 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and ACIS International Workshop on Self-Assembling Wireless Network, pp. 32–42. IEEE (2005)
16. Brooks Jr., F.P.: No silver bullet: essence and accidents of software engineering. IEEE Comput. (1987)
17. Eriksson, H.-E., Penker, M.: Business Modeling with UML: Business Patterns at Work. Wiley (2000)
18. Kitchenham, B.A., Pickard, L., Linkman, S., Jones, P.: A framework for evaluating a software bidding model. Inf. Softw. Technol. **47**(11), 747–760 (2005)

19. Jalote, P.A.: A Concise Introduction to Software Engineering. Springer Science & Business Media, London (2008). https://doi.org/10.1007/978-1-84800-302-6
20. Mili, H., Tremblay, G., Jaoude, G.B., Lefebvre, É., Elabed, L., Boussaidi, G.E.: Business process modeling languages: sorting through the alphabet soup. ACM Comput. Surv. (CSUR) **43**(1), 4 (2010)
21. Bider, I.: State-oriented business process modeling: principles, theory and practice (2002)
22. Shishkov, B., Dietz, J.L.: Deriving use cases from business processes, the advantages of demo. In: ICEIS, vol. 3, pp. 138–146 (2003)