Utilizing Business Process Models to Generate Software Test Cases

Sarah Khader[1] and Rana Yousef[2]

**[1] Department of computer information systems, University of Jordan
Amman, Jordan**

**[2] Department of computer information systems, University of Jordan
Amman, Jordan**

Abstract

A business process model (BPM) represents the step-by-step activities used to accomplish a business objective for the organization. The correctness of the BPM directly controls the correctness of the final developed software system. This research aims to utilize the BPM in the software development lifecycle in its very early phases, not only to provide a better understanding of the business case but to generate test cases from these models to help improve software testability. In order to accomplish this goal, a framework was developed to automatically generate test cases from business process models before any development has been conducted yet, emphasizing the principles of test-driven approaches for software development. Evaluation has shown that test case generation is possible using BPMs, In addition, a considerable part of the test cases generation process can be automatically compared to the traditional approach of producing test cases from the requirements documents. This in turn can improve and simplify software testability, and hence the overall development process, in this research the generated test cases set from the framework were compared to a test cases set generated from the requirements traditionally, results of the comparison were in favor of the framework in terms of time needed to generate test cases, Completeness, code coverage, productivity and test case affectivity.

Keywords: Business Process modeling; test case; BPMN.

1. Introduction

Due to the swift development of the internet, the competition in the world of developing software increased, and the need to have high quality software is now more important than ever. So what is high-quality software? It's not about the application that's the most publicized, or the fastest, most featureful, or best, often it has competing applications that are superior in many aspects. But at the end of the day, it is one of the most popular programs out there and is what many people like to use, what they find as the right tool for the job, and it simply works as expected.

The context within the software under development will certainly impact whether the software is considered high quality or not. A software crash in a standalone personal computer application may be irritating but a software crash in an industrial control system could kill someone [25]. In both cases, robustness is still a desirable and an important quality characteristic for the software. What is different is the level of robustness in the two different contexts. Therefore, while the context may determine the relative level of a particular characteristic, the quality characteristics and attributes remain the same.

In order to achieve high quality software more effort and concentration shall be spent in achieving a certain level of quality attributes this in turn requires - spending major effort and cost in the testing phase.

As known; According to a survey of IT executives performed by the Standish Group [1], only 39% of software projects succeeded, while 18% were challenged and 43% completely failed as shown in Figure 1-1.

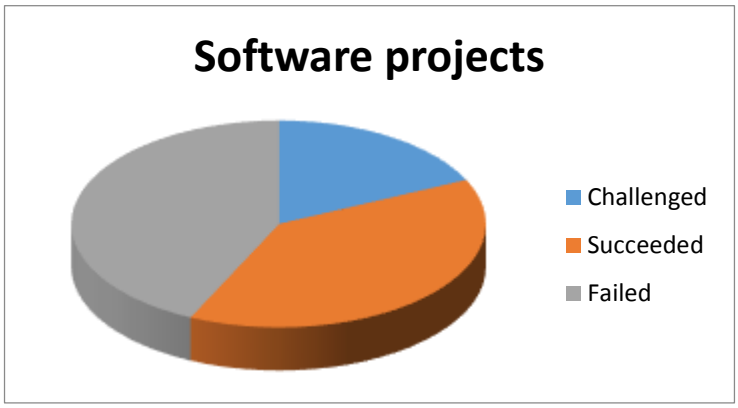

Fig. 1 Standish Group Report 2012

Furthermore, they say that an average software project runs 222% late, 189% over budget and delivers only 61% of the specified functions.

To fulfill the need to develop high quality and rapid software, a group of software developers met at the

Snowbird resort in Utah to discuss new lightweight development methods. They published the Manifesto for Agile in 2001 [26]; Agile software development is multiple software development methods in which solutions develop through merging between self-organizing and cross-functional teams. It encouraged adaptive planning, early delivery, evolutionary development, continuous improvement, and prompts fast and flexible response to change [27].

As organizations are moving towards being more agile, the need to bridge the gap between business and IT is increasing, and bearing in mind the increasingly competitive business world, companies can't adapt their structures and processes with the technological changes, Many of these companies decided to adopt IT technologies that supports modern business practices.

Agile development methods made maintaining high quality software even more challenging as features are developed on the fly and changes are continually imposed during the development process.

In order to maintain high quality rapid software, test planning, test case design and black-box tests should play a crucial role. Test cases are usually extracted in the requirement and specification phase. However, in this research, test cases are extracted as early as the business process modeling phase, which we expect to speed up the process due to utilizing available assets in an organization.

The rest of this paper consist of 4 subsections, first is the literature review where similar work for this research will be addressed, secondly the proposed framework where the research methodology for this will be presented, in the third section the experiment results and finally the last section will briefly discuss the thesis structure.

2. Literature review

This research will utilize the business process modeling quality measures in the very early phases of development process, in order to improve the testability of the software system through a semi-automatic generation of test cases.

There is plenty of literature which emphasizes the power and importance of BPM in the software development lifecycle. For example Joseph Barjis [4] in his paper introduced an innovative method for business process modeling for the purpose of software system design; the author has introduced some non-conventional frameworks to use with modeling.

Herden et al [5] proposed a new agile development methodology based on the business process modeling notation (BPMN). Their methodology defines an agile process which uses this notation to generate prototypes from the early stages of the development process, easing the validation of software by the customer.

David J. Teece [11] in his paper discussed the importance of business models and the relations between business models and business strategy.

Charles Baden-Fuller and Mary S. Morgan [20] show in their paper that studying business models as models is very rewarding as they can be treated as an exemplar models that can be copied or could be used as short hand description to scale models.

Shafer, Smith & Linder [21], in their paper "The power of business models" emphasized the growing importance of business process modeling from different angels, Different definitions were given to describe the process of business modeling, and the authors also discussed some of the business process modeling problems.

The standardization of business process modeling notation, especially 2.0 by OMG gave opportunity to the emergence of tools for modeling and execution of business processes that helped fast prototyping of applications. Such as the work of Herden [5] in their paper the authors propose a new agile development methodology based on BPMN that defines an agile process which uses BPMN to generate prototypes from the early stages of the development process, easing the validation of software by the customer. Following the approach of agile methods that prioritize and produce executable artifacts rather than textual descriptions that increases the time required for system development. Furthermore, the graphical representation of BPMN, used to specify the flow of tasks and services, giving scope for users to play the role of business analysts.

With business process modeling, companies and organizations can gain explicit control over their processes. Currently, there are many notations in the area of business process modeling, where Business Process Model and Notation (BPMN) is denoted as the standard as Kocbek [22] provided in their paper the state-of-the-art results addressing the acceptance of BPMN, while also examining the purposes of its usage. Furthermore, the advantages, disadvantages and other interests related to BPMN were also investigated to achieve these objectives, a Systematic Literature Review (SLR) and a semantic examination of articles' citations was conducted.

In this research the BPMN was used for its powerful XML based structure, as business process models presented in BPMN based format are the input for the framework explained in details in chapter three, the xml file is parsed easily using XPath, However several updates and changes are proposed to enhance BPMN, one of these changes is what Giacomo [24] proposed in their paper that traditional business process modeling notations, including the standard Business Process Model and Notation (BPMN), rely on an imperative standard wherein the process model captures all allowed activity flows. In other words, every

flow that is not specified is implicitly disallowed. Recently, several researchers have exposed the limitations of this standard in the context of business processes with high variability. As an alternative, declarative process modeling notations have been proposed. It has been recognized that the boundary between imperative and declarative process modeling is not hard. Instead, mixtures of declarative and imperative process modeling styles are sometimes preferable, leading to proposals for hybrid process modeling notations. These developments raise the question of whether completely new notations are needed to support hybrid process modeling. Their paper answers that question negatively. The paper presents a conservative extension of BPMN for declarative process modeling, namely BPMN-D, and shows that Declare models can be transformed into readable BPMN-D models.

As can be seen from the above literature, most of the research in utilizing BPM in the software development process is focused on generating prototypes and/or providing methodologies to understand and extract requirements for the system. However, our work in this research is concerned with generating test cases.

There is remarkable amount of literature on methods for generating test cases, however few is available on generating test cases from business process models, and the researches that do, only describe models for specific languages, our method on the other hand would work with any business process model described in the standards of OMG for BPMN 2.0.

Guangquan, Mei and Jun [6] in their paper produced test cases for web services based on extended UML activity diagram; they propose a method that produces relevant test cases only.

Fanjul, Tuya and Riva [24] proposed generating test cases from BPEL using systematic procedure; however none of the methods above uses CPM; CPM is a pragmatic representation of the well-known equivalence partitioning method used in testing (The Testing Standards Working Party, 1998) which is a technique that separates all input data (testable data) of a software into partitions of equivalent data from which test cases can be derived, giving the pragmatic nature of CPM, utilizing CPM in test case generating will help speed up the process.

One of the methods used in this research is the category partition method(CPM), CPM is mainly a method for extracting test cases from high system specifications documents, in this research it was used to extract test cases from the business process models automatically, one of the main corners in literature review is the 1988 paper for Ostrand and Balcer [7] in which they have introduced the category partition method, according to the authors The advantages of this method is that it gives the tester the possibility to modify the test specification when necessary, and the ability to control the complexity of the tests generated.

Enhancement to the CPM method is a research area in which researches like Liu [8] are interested, in their paper the authors produced a new framework with the name (CHOC'LATE) this framework was presented to support their argument that The traditional approach to partitioning the input domain may not be adequately strong to ensure similar execution actions for the same resultant partitions. Instead they propose that output scenarios should also be explicitly considered for any partition testing method to improve the homogeneity of the input partitions, which, in turn, is the key factor for high fault-detection effectiveness.

3. The proposed Framework

One of the most commonly used techniques in software testing is equivalence partitioning [9] which is a technique that separates all input data (testable data) of a software into partitions of equivalent data from which test cases can be derived. Each partition will be presented by well-designed test cases that cover that particular partition, this technique reduces the time and number of test cases required to test software by producing test cases that expose classes of errors.

Equivalence partitioning is a well-practiced technique and is also very pragmatic, on the other hand it is seldom described in a formal or precise way, probably it is best presented by the Ostrand and Balcer's Category Partition Method (CPM) [7]. In our research the category partition method is used to generate test cases from high level software specifications (Business process models), the CPM method is specialized to generate test frames automatically from business process models based on all possible paths generated from the model, the resulting test frames can be used to generate the final test cases.

Figure 1 illustrates the five main phases which were developed this in research in order to extract test cases from business process models.

Step 1: Obtain the business process model and test it

In this research the "Innovator for business analysts"[1] tool is used to plot and test all business models, other tools can be used as well if complies with the OMG standards for business modeling notation 2.0.

The generated BPM is tested against:

- Notation violations: These are predefined by the "Innovator for business analysts" tool, and we

[1] Open source tool: http://www.mid.de/en

used the predefined list as is, for example if two opposite gates are sequent, the system will automatically display an alert.

- Unresolved model references: The model is validated against Nil or unresolved references.
- Configuration violations: These configurations are predefined by the user and the model will be verified against those, this feature will be used to apply some constraints to some business process models.

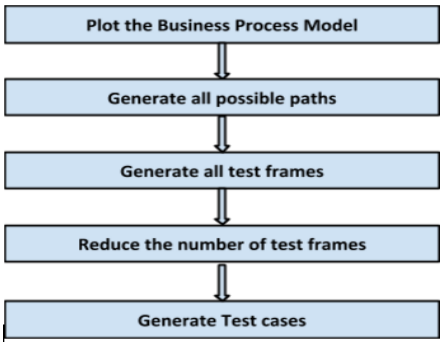

Fig. 2 The proposed framework

Step2: Generate all possible paths

To generate all possible paths for a model from the start event to the end event, the concept of depth first search is used [10], all tasks and diagram activities are considered as nodes, all sequence flows in the models are considered as edges and the root node is the start event for the model.

The algorithm will start the navigation from the root node and adds all activities as nodes in the path to form a possible path until a gateway is encountered, in this research all model gateways are categorized under (AND, XOR,OR and parallel gateways) and the resulting paths are treated as follows:

- In the case of AND, all branches shall be executed in parallel.
- In the case of XOR, exactly one branch will be executed.
- In the case of OR and PARALLEL gateways, one or more of the branches shall be executed in parallel.

Step3: Generate all possible test frames

A test frame in this research is used in the context of an extended view of the processes in the model with knowledge about their inputs, so test frames will bind partitions to categories as will be explained below.

The CPM method was adapted in this research to automatically generate high level test frames based on possible paths which are determined by business rules and constraints in the business process model. The generated test frames can then be used, after filtration and specialization, to create the actual test cases.

The Ostrand and Balcer's Category Partition Method (CPM) was adapted in this research to perform the follows steps:

a. Identify independently testable features: testable features are the set of all inputs to a process.

b. Identify categories: which are the characteristics of each input element.

c. Partition categories into choices: where choices are the interesting cases of the sub domain.

Step 4: Reduce the number of test frames

Combining the choices generated from the previous step would result in a tremendous amount of test frames, so constraints will be defined to reduce the number of test frames and to eliminate the meaningless combinations, these choices are provided manually by the test engineer and they must be one of the following:

- The if property: If the length of the input string is zero, this means that it is not a special character. So, for the special character choice we put (if! Zerovalue).
- Error property: We exclude erroneous situations for example if the input size is less than zero then we will consider it only once and we mark it with the Error property.
- Single property: It is used as the error property but with a different meaning. When we use it we indicate that we want to use this choice only in one combination and not in multiple combinations.

Step 5: Generate test cases

After generating test frames a simple instantiation shall be performed to produce test cases.

4. Experimental data and results

Most test case generation methods are usually applied after the development process is underway, and it relays

either on the requirements' document or the generated software. In this section, we are going to compare the test cases generated using our developed framework with those generated using a traditional requirements-based method in terms of completeness, time, defect detection, test case affectivity and test case generation productivity.

In order to perform the comparison, we used a real case study of a software system developed for the deanship of academic research (DAR) in the University of Jordan; The DAR Employees' Evaluation system. This system was developed to be used to track and log employees' accomplishments. It helps users logging their achievement for the day, and help admins manage and monitor performance.

After careful analysis of the DAR system requirements the following BPM model was obtained.

As can be seen from Figure 2, the system consists of three main processes: The admin system definition process, the admin reporting process and the user process.

PROCESS 1- Admin System Definition Process: In this process the system administrator defines all system sections, procedures and users.

PROCESS 2- Admin Reporting Process: In this process, the admin can view and generate reports.

PROCESS 3- User Process: In this process, the user can view his/her procedures, log in new procedure whether it is an urgent procedure or a regular one.

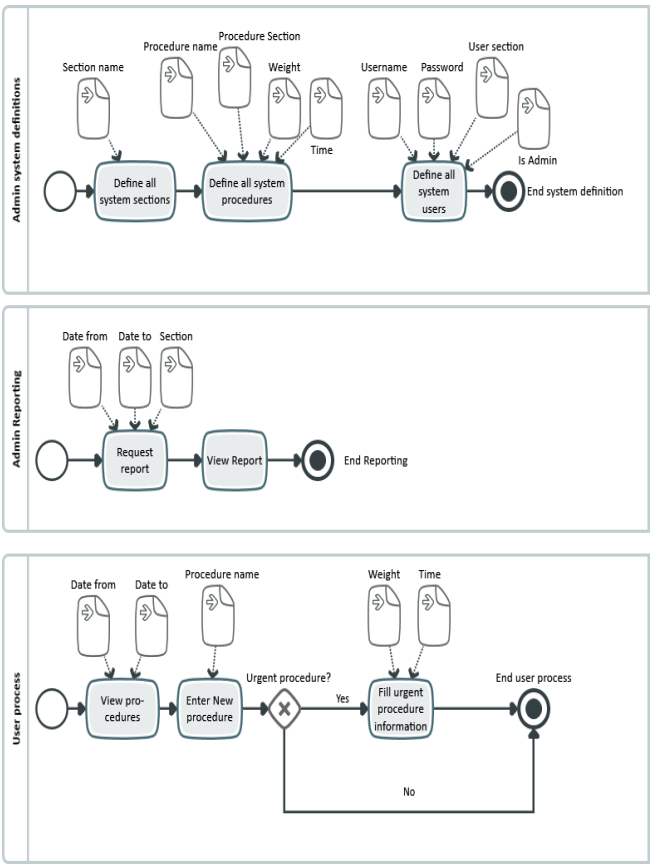

Fig. 2: DAR Business process model

A sample of the generated test cases is shown in Figure 3 for the reporting process.

For the purpose of comparing our generated test cases produced by the framework, we asked a professional quality assurance team leader to generate test cases for this project using the traditional requirements-based method. Then, we compared test cases generated using both methods in terms of:

1. Completeness: the number of generated test cases
2. Time: required time to generate test cases
3. Defect detection: this is the measure of test case ability to uncover defects.
4. Test case affectivity: the ratio of the numbers of errors detected to the number of test cases executed.

Affectivity = (Errors detected / test cases executed)

5. Test case generation productivity: another important measure is the productivity of test cases generation which is reflected on project time, it is the ratio of number of test cases produced by minute. Productivity= Test Cases/Tester Days

Test frame number: 1
Date from - Value = Empty
Request report
View Report

Test frame number: 2
Date from - Value = More than Date to
Request report
View Report

Test frame number: 3
Date from - Format = Not in Date format
Request report
View Report

Test frame number: 4
Date to - Format = Not in Date format
Request report
View Report

Test frame number: 5
Date to - Value = Empty
Request report
View Report

Test frame number: 6
Date to - Value = Less than Date from
Request report

Fig. 3: A sample of test cases generated by our framework: BPMN-based framework

Table 1 below shows the results of comparing the test cases generated by our framework and those generated by the traditional requirements-based method.

Table 1: Comparison results

Criteria	Requirements-based	BPMN based
Completeness	14	45
Time	one working day	30 minutes
Defect detection	4	14
Test case affectivity	29%	31%
Code coverage	23%	86%
Test case generation productivity	0.03 test case per minute	90 test cases per minute

To sum it up, BPMN based framework generated more test cases therefore more complete and test cases were generated in less time than the traditional method which led to better test case generation productivity.

Also, as shown in table 1 above, BPMN-based framework produced better results in defect detection. As the test case affectivity measure depends on two variables which are the number of errors detected and the number of test cases executed, a relatively close result in this criteria were figured out, with an improvement by 2%.

5. Conclusions

Our main objective in this research was to improve software testing through utilizing business process models in generating software test cases.

Business process modeling was used at an early stage in the software development process to help produce test cases automatically with an ultimate goal to enhance the software development process.

The new method for generating test cases is based on BPMN 2.0 OMG standards, the BPMN xml file is parsed using XPath technique, then all possible paths in the business process model were extracted using depth first algorithm while bearing in mind different model gateways and different gateway complexities. After producing all paths from the diagram, the test engineer inputs all required data for the CPM method (categories and partitions) for each process input for each process, and the framework will produce and combine test cases to be used later in the development process.

As a method for evaluation, we have studied and compared the behavior of test cases generated using traditional ways after the development process is underway, and test cases generated from the BPMN-based framework at an early stage before the development process starts, different comparison criteria were evaluated; completeness, time, defect detection, Test case affectivity and test case generation productivity. Results showed that our framework has improved test case generation compared to the traditional requirements-based method, where the number of test cases has been improved by a factor of 3.2, 62.5% reduction in time, 3.5 in defect detection ability, 2% improvement in test case affectivity.

Although our framework can be applied to plan-driven approaches, where having a preliminary tests at early stages of development is an added value, it would be difficult to deploy our framework for large systems, due to the complexity limitations.

In this context we will briefly describe some interesting research topics, which worth further investigation: Improvement to the algorithm complexity can be considered; although this method is aimed towards small to mid-size projects it is crucial to make some improvement to the time complexity especially if the framework is used to generate test cases for larger projects. Also, generating automation scripts to be used directly with automation tools would be very helpful, it is possible to generate test scripts from this method and use them as input directly to a test automation tool. Finally, improvement to CPM stage of the method could be done by building a database for all entered categories and partitions in the system to be used in later projects.

References

[1] Standish, the Standish Group International, Inc. (2012).Third Quarter Research Report, West Yarmouth, MA, technology. Harvard business school press, Boston.

[2] Somerville, I. (2010). **Software engineering**. 9th ed. Reading, MA: Addison-Wesley

[3] Pravin, A., & Srinivasan, S. (2013). **Effective Test Case Selection and Prioritization in Regression Testing**. Journal of Computer Science, 9(5), 654.

[4] Barjis, J. (2008). The importance of business process modeling in software systems design. Science of Computer Programming, 71(1), 73-87.

[5] Herden, A., Farias, P. P. M., de Andrade, P. R. M., & Albuquerque, A. B. (2014). **Agile PDD–One Aproach to Software Development Using BPMN**. In11th International Conference Applied Computing, Porto, Portugal.

[6] Guangquan, Z., Mei, R., & Jun, Z. (2007).**A business process of web services testing method based on uml 2.0 activity diagram**. In Intelligent Information Technology Application, Workshop on (pp. 59-65). IEEE.

[7] Ostrand, T. J., & Balcer, M. J. (1988). **The category-partition method for specifying and generating functional tests**. Communications of the ACM, 31(6), 676-686.

[8] Liu, H., Poon, P. L., & Chen, T. Y. (2015). **Enhancing partition testing through output variation**. In Proceedings of the 37th International Conference on Software Engineering-Volume 2 (pp. 805-806). IEEE Press.

[9] Wiegers, K.(1999). Software Requirements. Microsoft press, Redmond work. John Wiley & Sons, Inc,

[10] Even, S. (2011). Graph algorithms. Cambridge University Press.

[11] Teece, D. J. (2010). **Business models, business strategy and innovation**. Long range planning, 43(2), 172-194.

[12] Sánchez-González, L., García, F., Ruiz, F., & Mendling, J. (2012). **Quality indicators for business process models from a gateway complexity perspective**. Information and Software Technology, 54(11), 1159-1174.

[13] Pravin, A., & Srinivasan, S. (2013). **Effective Test Case Selection and Prioritization in Regression Testing**. Journal of Computer Science, 9(5), 654.

[14] Mendling, J.(2008). Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness: Springer Publishing Company, Incorporated.

[15] Khlif, W., Zaaboub, N., & Ben-Abdallah, H. (2010). **Coupling metrics for business process modeling**. International Journal of Computers, 4(4).

[16] Baden-Fuller, C., & Morgan, M. S. (2010). **Business models as models**. Long Range Planning, 43(2), 156-171.

[17] Van der Aalst, W. M. (2007). Trends in business process analysis. In Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS) 2007 (pp. 12-22).

[18] Vanderfeesten, I., Cardoso, J., Mendling, J., Reijers, H. A., & van der Aalst, W. (2007). Quality metrics for business process models. BPM and Workflow handbook, 144, 179-190.

[19] Younessi, H. (2002). Object Oriented Defect Management of Software. Prentice Hall PTR.

[20] Baden-Fuller, C., & Morgan, M. S. (2010). **Business models as models**. Long Range Planning, 43(2), 156-171.

[21] Shafer, S. M., Smith, H. J., & Linder, J. C. (2005). The power of business models. Business horizons, 48(3), 199-207.

[22] Potrč, T., Baumgartner, S., Roškar, R., Planinšek, O., Lavrič, Z., Kristl, J., & Kocbek, P. (2015). Electrospun polycaprolactone nanofibers as a potential oromucosal delivery system for poorly water-soluble drugs. European Journal of Pharmaceutical Sciences, 75, 101-113.

[23] De Giacomo, G., Dumas, M., Maggi, F. M., & Montali, M. (2015, June). **Declarative Process Modeling in BPMN**. In Advanced Information Systems Engineering (pp. 84-100). Springer International Publishing.

[24] García-Fanjul, J., Tuya, J., & De La Riva, C. (2006). Generating test cases specifications for BPEL compositions of web services using SPIN.International Workshop on Web Services–Modeling and Testing (WS-MaTe 2006) (p. 83)

[25] Meyer ,Bertrand.(1997) **Object-Oriented Software Construction**,Prentice Hall; 2 edition

[26] Kent Beck. (2001). **Manifesto for Agile Software Development**. [ONLINE] Available at:http://agilemanifesto.org/. [Accessed 13 October 15].

[27] **Agile alliance** (2013). [ONLINE] Available at: http://www.agilealliance.org/the-alliance/what-is-agile/. [Accessed 13 October 15]

[28]