

# Prioritization of Re-Executable Test Cases in Model Based Environment- A Survey

Komal Jadhav<sup>1</sup>, Sharmila Shinde<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, Savitribai Phule Pune University, Pune

<sup>2</sup>Jaywantrao Sawant College of Engineering, Hadapsar, Pune, Satavnagar, Handevadi Road, Pune-28, India

**Abstract**— In case of regression testing and prioritization approaches, specification level design and requirements concerns motivate changes in the source code. Traditional approaches are bottom up(or white box) and do not leverage on these concerns, so the new concept which considers these concerns is developed called as model based testing approach. Model-based testing approaches supports a top-down (or black box) testing approach, where design and requirements models are used in support of test generation. They enhance code-based approaches with the ability to test from a higher-level design and requirements perspective. A model-based regression testing and prioritization approach is presented that efficiently selects test cases for regression testing based on different concerns. In particular we describe how to convert a UML diagram to xml file, how to derive the re-executable and reusable test cases and how to support concern- based regression testing and prioritize the re-executable test cases. Efforts have been initiated to map UML diagrams to XML documents, because Unified Modeling Language (UML) is a standard object oriented design modeling language for business and technical systems and Extensible Markup Language Schema (XML Schema) provide a means for defining the structure, content and semantics of XML documents and XML is widely being accepted as information representation and sharing language across Internet. In order to build a complete system, we have to map the static as well as the dynamic aspects of UML Diagrams. After this conversion, xml file is used to generate test cases which are reusable and re-executable. Re-executable test cases are further prioritized based on user concerns or dependency of re-executable test cases, which can be identified by activity diagram. This will minimize the efforts needed to validate new software versions and it will improve the overall productivity of the software development process.

**Keywords**— activity diagram, regression testing, test prioritization, reusable test cases, re-executable test cases.

## I. INTRODUCTION

This document is a template.

With the advent of technology software becomes very crucial part in all the institute and industries. To develop particular software and to test it as per customer requirement is very important because the software which is not able to satisfy customer requirement after development will lead to increase and waste of cost, time and effort of all parts of organization. Model-based test case generation is gaining acceptance to the software practitioners. Advantages of this are the early detection of faults, reducing software development time etc. UML(Unified Modeling Language) is used to develop software designs, which includes various diagrams for complete

representation of the software. XML (Extended Markup Language) is widely used for information exchange on internet and to define structures and content using user defined tags. Few work on the test case generation using activity diagrams is reported in literatures. To increase the productivity better test case suit generation is very important for any small or large software application. To generate better test cases of any project which reduce number of test cases to be executed and also all part coverage is necessary now a days. Regression testing is used for generating the test cases, and it allows to reduce the test cases in case of modifications in the system. So, the idea to generate and prioritize optimum number of test case suite for better utilization of resources motivates to develop this system.

Model-driven software development is a new software development paradigm. Its advantages are the increased productivity with support for visualizing domains like business domain, problem domain, solution domain and generation of implementation artifacts. In the model-driven software development, practitioners also use the design model for testing software specially object-oriented programs. Three main reasons for using design model in object-oriented program testing are:

- Traditional software testing techniques consider only static view of code which is not sufficient for testing dynamic behavior of object-oriented system.
- Use of code to test an object-oriented system is complex and tedious task. In contrast, models help software testers to understand systems better way and test information only after simple processing of models compared to code.
- Model-based test case generation can be planned at an early stage of the software development life cycle, allowing to carry out coding and testing in parallel.

For these three major reasons, model based test case generation methodology becomes an obvious choice in software industries. Activity diagram is an important diagram used for business modeling, control and object modeling, complex operation modeling etc. Main advantage of this model is its simplicity and ease of understanding the flow of logic of the system.

Regression testing allows to optimize and reduce test cases, so that there are minimum number of test cases to be generated after some changes in the design of software during software development cycle.

## II. LITERATURE SURVEY

In [1], Roberto S. Silva Filho, Christof J. Budnik, William M. Hasling, Monica McKenna and Rajesh Subramanyam have proposed a model based regression testing and prioritization scheme which efficiently selects test cases for regression testing based on different user defined concerns. It depends on traceability links between models, test cases and code and user defined properties associated to model elements. Here, an automatic tool called TDE/UML is used which generates test cases using UML diagrams and categorizes them as reusable and re-executable test cases and prioritizes them. The proposed approach in this paper is top-down approach as compared to traditional code based bottom-up approaches, because it works along with the software development life cycle i.e from starting stage of the software. This approach works efficiently, as it detects the errors in the early development stages of the software. The technique reduces the efforts needed to validate the new versions of the software and improves the overall productivity of the software.

Model-based regression testing ensures the reliability of the evolving softwares by optimally selecting the test cases to test the affected portion of the software. This technique promises the reduction in labour, time and cost to test the new version of the software.

In[2], Mr. Rohit N. Devikar had presented the automatic tool, Model-based regression testing tool(MBRT), which is java based tool and used to reduce, generate and also categorize test cases as obsolete, reusable and re-testable test cases. In this paper class diagram and state machine diagram are used for regression testing and flow graph is used to generate the test cases.

In [3], to optimize the priority of the test cases at different points in the design cycle, tool called Echelon is developed by A. Srivastava and J. Thiagarajan, which is a test prioritization system, that prioritizes the set of test cases of any application, based on changes made to the program. Echelon uses a binary matching system that computes the differences at a basic block granularity between two versions of the program in binary form. Echelon works on heuristic approach. Echelon runs under the Windows environment.

Features of Echelon includes following:

1. Echelon prioritizes tests into an ordered sequence of tests based on program change.
2. Echelon compute changes between programs at a very fine granularity—at the level of basic blocks— using an accurate binary matching algorithm.
3. Echelon uses a fast, simple algorithm that works well in practice and does not attempt an expensive data flow algorithm to determine which new code will be covered by which existing tests.
4. Echelon operates at the binary level making it easier to integrate into the development process. Echelon scales to real product binaries in large-scale development environments.
5. Echelon also produces a list of program source code that will not be covered by any of the given

existing tests. New tests will be needed to test them.

In [4], R. France and B. Rumpe have given an overview of current research in Model Driven Engineering (MDE). The research work in this paper is focused on providing technologies that address the recurring problem of bridging the problem-implementation gap. We also encourage research on the use of runtime models. In this paper, a vision of MDE environment is presented, that if realized, can result in improvement in software development productivity and quality. Progressively closer approximations of the vision will have increasingly significant effects on the effort required to develop complex software. The vision can act as a point of reference against which MDE research progress can be informally assessed.

In [5], a methodology and tool is presented by L. C. Briand, Y. Labiche, and S. He that support test selection from regression test suites based on change analysis in object-oriented designs. Regression test cases are categorized as Reusable, Retestable, and Obsolete.

This paper focuses on automating the regression test selection based on design model represented by Unified Modeling Language(UML) and the traceability linking the design to test cases. UML is used in this case as it is becoming industry de-facto standard.

Motivations behind using test case selection techniques at design level are:

- Estimation of the extent requiring regression testing earlier on, at the end of the design of the new system version.
- Programming language independent regression test tools, based on UML.
- Storage and updation of dependencies between test cases and code statements to find traceability links between code and test cases.
- Dynamic analysis for identifying possible dynamic bindings between methods at run-time. Using UML designs enables the easy retrieval of relevant static and dynamic information (e.g., class interactions at run-time from sequence diagrams).

There are certain drawbacks of the system:

- Requires the designs to be complete and up-to-date.
- Not easy in practice.
- Some (potentially faulty) changes to the source code may not be detectable from UML documents.

In[6], to remove disadvantages of code based regression test selection, a new specification based test selection technique is developed by Y. Chen, R. L. Probert, and D. P. Sims, which is based on customer requirements. The basic model used is Activity diagram which is part of UML(Unified Modeling Language). In this paper two types of regression tests are selected as targeted tests and safety tests. Targeted tests ensure that important current customer features are still supported in the new release. Safety Tests are risk- directed, and ensure that potential problem areas are properly handled. Proposed test selection technique is based on a practical risk analysis model.

### III. PROPOSED WORK

Various Techniques have been proposed to prioritize the test cases in model driven environment. In the proposed approach, Activity Diagram will be used to generate the test cases for the system and prioritize them. Heuristic based approach will be used to prioritize the test cases. Activity diagram will be first converted into xml format, the conversion is carried out to generate test cases from xml file. A basic design's xml format is first saved and after changes to the basic design of the activity diagram, again activity diagram to xml conversion is carried out. Differences between the two xml files, old and new one will be calculated and test cases are categorized as reusable and re-executable test cases. Reusable test cases are common and re-executable test cases corresponds to changed part of the design. Regression testing is applied to generate the test cases, but as compared to the traditional approach the duplication is removed by categorization of test cases. After this categorization, re-executable test cases will be prioritized using the heuristic approaches. Pacestar UML Diagrammer is used to generate the test cases and its design structure saved in text format is used for conversion of Activity diagram to xml file. Prioritization of re-executable test cases will reduce the test cases will reduce the testing time, efforts and cost of the software and will improve the overall productivity of the software by reducing the efforts needed to validate the new versions of the software. Figures 1 and 2 show the Pacestar UML Diagrammer tool and the text format of the activity diagram respectively.

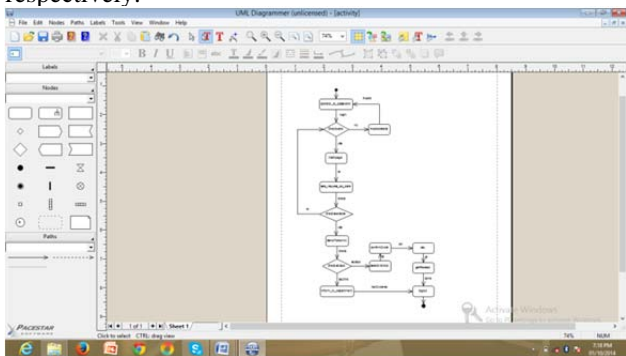


Fig 1: Pacestar UML Diagrammer Tool to create Activity Diagram

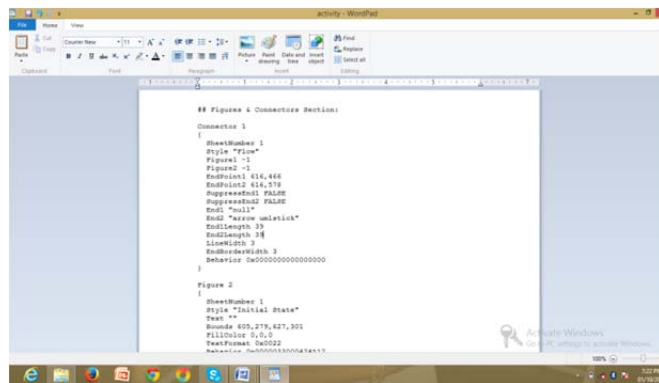


Fig 2: Activity Diagram in text format

### IV. CONCLUSIONS

In software development life cycle, software testing is very important part, as the success of the software project depends on it.

It is therefore very essential to handle this task of software development very carefully. In our paper, we present a regression testing and prioritization technique which will work from early stages of the software development life cycle and will reduce time, efforts and cost for testing. Heuristic approach will be used to prioritize the re-executable test cases which will consider various aspects for prioritization as user requirements, dependency of re-executable tests identified by activity diagram. This prioritization technique will result in better improved version of the software with minimum efforts and will increase software productivity. Test cases can be further prioritized by considering various domains like finance, education, medical, etc.

### ACKNOWLEDGMENT

The heading of the Acknowledgment section and the References section must not be numbered.

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the people who make it possible. Ms. H.J. Thanki author wish to thank honourable guide and H.O.D. Prof. S.M.Shinde, PG coordinator Prof. M.D. Ingle and Principal Dr. M.G. Jadhav of JSPMs Jaywantrao Sawant College of Engineering, Hadapsar, Pune, for providing vital comments, information, and review of this paper. We also acknowledge thanking different sites and references which we receive from different sources.

### REFERENCES

- [1] Roberto S. Silva Filho, Christof J. Budnik, William M. Hasling, Monica McKenna, Rajesh Subramanyam, "Supporting Concern-Based Regression Testing and Prioritization in a Model-Driven Environment", 34<sup>th</sup> Annual IEEE Computer Software and Applications Conference Workshops, 2010.
- [2] Mr. Rohit N. Devikar, Prof. Manjushree D. Laddha, "Automation of Model-based Regression Testing", International Journal of Scientific and Research Publications, Volume 2, Issue 12, December 2012.
- [3] A. Srivastava and J. Thiagarajan, "Effectively Prioritizing Tests in Development Environment," in Intl. Symposium on Software Testing and Analysis Roma, Italy: 2002..
- [4] R. France and B. Rumpe, "Model-driven Development of Complex Software: A Research Roadmap," in Future of Software Engineering: IEEE Computer Society, 2007.
- [5] L. C. Briand, Y. Labiche, and S. He, "Automating regression test selection based on UML designs," Inf. Softw. Technol., vol. 51, pp. 16-30, 2009
- [6] Y. Chen, R. L. Probert, and D. P. Sims, "Specification-based Regression Test Selection with Risk Analysis," in 2002 Conference of the Centre for Advanced Studies on Collaborative Research Toronto, Ontario, Canada: IBM Press, 2002.