A Survey of UML-Based approaches to Testing

Swati Tahiliani[1,] **Pallavi Pandit**[2]

[1] Medi-Caps Institute of Technology and Management,

[2] Medi-Caps Institute of Technology and Management

Abstract:

Apart from application modeling, the Unified Modeling Language (UML) is also used for designing the tests on various levels (unit, integration, system tests). Designing tests based on UML diagrams will give high quality results, a more robust system. The advantage being that, as same set of UML diagrams are used for modeling production code, same can be used for automated testing. This would enable the developers and testers to test even at the early stages of software development. Different models can be used for defining tests and production code in many ways. In this paper, we will explore different algorithms and techniques applied on use cases and other UML diagrams and cite the work done in the respective areas.

Keywords: Automated Testing, Model Based Testing, test cases, test case generation, techniques for deriving test cases, UML, use cases.

1. Introduction

Many software products fail in some or the other ways, just because of improper testing. Although the software is tested, but testing starts in the later phases of the software development process. A small flaw in analysis or design becomes as a big problem at deployment. Some of these problems can be avoided, if one follows test-first approach in software development process. Our paper focuses on the same issue. In the first part of our paper, we have discussed use cases, their utility, testing approaches, related surveys and application areas. In the second half of our paper, we have enlisted the approaches used for testing, using various UML diagrams.

2. Use cases

2.1. Introduction to use cases

A use case is a description of a set of actions that are performed by the system to produce certain output. Its emphasis is on 'what' is expected from a system and not 'how' that can be achieved. An actor may be any user of use cases, who expects the system to show a particular behavior. A use case describes this behavior. Or we can say that, it is a collection of the actions that are performed to show this behavior.

2.2. Usage of use cases

(a) Use cases enable the analyst(s), designers, developers and the intended users of the system to arrive at common grounds regarding the requirements.

(b) Use cases also play a major role in validating the system architecture.

(c) Use cases allow us to visualize, document, code and test the requirements of a system.

2.3. Taking care while writing use cases

Use cases have certain limitations, as summarized below[1]:

(a) Using use case templates correctly is a skill required by the writer of the use case.

(b) Proper and accurate interpretation of use cases is a difficult task for both developers and end users.

(c) Extreme Programming and Agile modeling approaches prefer user stories to use cases, considering use cases to be too document-centric.

(d) Use case developers cannot decide whether to incorporate GUI details in a use case and to what degree.

(e) Not all systems are best described using functionality-driven approach followed by use cases.

Other drawbacks encountered while writing use cases include functional decomposition, which are described in detail in [2].

3. Use Cases to Test Cases

3.1. Use cases as a basis for testing

Use cases are mainly used to specify the required functionality of an object oriented system. They are used to serve many other purposes too, for ex. providing prototype test cases. Deriving test cases from use cases facilitates verification testing (the software should conform to its specifications), and are also used in validation testing (the software should do what the user really requires).[3]

Use Cases represent functional requirements and functional requirements need to be tested.
So, the use case could be considered one of the test case because you just need to test that if the use case[4]:

1. Is not getting initiated when the pre-condition is false.
2. Is executing as specified.
3. Giving the expected results (post-condition).

Description of a use case involves both user steps and system steps. But its corresponding test cases contain only user steps, as the system steps will automatically be verified if the outcome of test case is correct.

3.2. Test cases derived in a different way

Use cases are used here in a different way to derive test generation. They are converted into sequences of so called action words, which are simple translation from the events listed in the use cases. For ex. An action making a phone call is translated to the sequence of key strokes that conduct the action on the system under test (SUT). The action word sequences are then processed by test coverage language, which will be used in a simple algorithm for test generation[5]. The selection of a particular test coverage language requires the fulfillment of criteria, as outlined in [6].

3.3. Automatic Test Generation

The approach[7] is to generate tests from a formalization of the requirements of a system, in the context of object-oriented embedded software. They defined a simulation model of the use cases. The simulation model is used to explicitly build a model of all the valid paths through use cases, and to extract relevant paths from it using coverage criteria. These relevant paths are 'test objectives'. Here the test cases are generated in two steps: The derivation of test objectives from the use cases constitutes the first step. The second step involves translation of test objectives into test scenarios.

4. Related Surveys Of Techniques For Deriving Test Cases From Use Cases

Extensive work of deriving test cases from use cases has been cited in [8]. Many other approaches for deriving test cases from use cases based on functional requirements have been listed in [9]. The authors in [10] suggest exploring variabilities in use cases based on software product families. Five black box testing methods have been compared by [11] for testing object-oriented software, including derivation of test cases from use cases. A systematic review of Model-Based Testing approaches is given in [12].

5. Use cases are used for performing the following kinds of tests

User acceptance tests, system integration tests, function verification tests, unit tests, [13], manual black-box testing, automated testing, component based testing[14], automated GUI testing[15].

6. Application areas for testing based on use cases

(a) Testing Web Applications[16]
(b) Assessment of the reliability of the system, by finding a reliability metric for each test case T.[17]
(c) Generating interaction test cases for mobile phone systems. [18]
(d) Using use cases, misuse cases and mitigation use cases to incorporate security characteristics into the software product early in the lifecycle[19]

7. Suitability of UML for testing

UML is intended for communication and modeling. In addition to this, it can be utilized for testing provided that there is not another existing more suitable technique.

Testing implies the conformance between requirements and implementation. Since UML is used for specifying the requirements, hence it is used to support testing.[20]

UML can be used throughout the phases from analysis to testing. The information which is not depicted in the models or the information which cannot be modeled using UML can be reverse engineered later on into UML from code or other sources.[21]

Limitations and future work of Model-based testing (MBT) are described in [22].

8. Test Case Generation from UML Models

Apart from use cases, several UML Diagrams have also been extensively used for the test case generation process as summarized in the table below:

Table1: Techniques used for deriving test cases from UML Diagrams

UML Diagram used	**Algorithm/Technique**	**Work done in this area**
Sequence Diagram	Labeled Transition Systems	Test Case Generation by means of UML Sequence Diagrams and Labeled Transition Systems by Emanuela G. Cartaxo[23]
Sequence Diagram	Enrich the sequence diagram with class diagram and OCL constraints information to construct a structured composite graph called SCG.	Automatic Test Data Synthesis using UML Sequence Diagrams by Ashalatha Nayak and Debasis SamantaI[24]
Sequence Diagram	OCL, functional predicates and function minimization technique.	Test Case Automate Generation from UML Sequence Diagram and OCL Expression by Li, Bao-Lin Li, Zhi-shu Qing, Li Chen, Yan-Hong[25]
Sequence Diagram	Genetic algorithm	Automated Test Cases Generation from UML Sequence Diagram by A.V.K. Shanthi and G. Mohan Kumar[26]
Class Diagram and Activity Diagram	Data Mining Approach	A.V.K. Shanthi, Dr.G.Mohan Kumar, Automated Test Case From UML Diagram Using Data Mining Approach by A.V.K. Shanthi and G. Mohan Kumar[27] Automated Test Cases Generation For Object Oriented Software by A.V.K. Shanthi, Dr.G.Mohan Kumar.[28]
Use case diagram and state transition diagrams	XML-based tool for model transformation	UML-Based Statistical Test Case Generation by Matthias Riebisch, Ilka Philippow and Marco Götze[29]
Express a use case as a UML activity diagram	Use the Category-Partition method [30] to generate test cases.	TDE/UML approach Taken from [31]
UML Activity and Sequence Diagrams	Use a behavioral model to depict the functionality expected in the system under test	[Labiche02] Taken from [31]
UML Activity Diagrams	Annotated the UML Activity Diagram with categories, choices and restriction.	[Ruder04] Taken from [31]
State Transition Diagram	AI Planning Techniques	Automated Test Case Generation from Dynamic Models by Peter Fröhlich and Johannes Link[32]
Activity Diagrams	Derivation of Domain Test Scenarios from Activity Diagrams.	[Reuys03], [Ruder04], [Ruder04-02] Taken from [31]
UML collaboration diagrams and statecharts	Intermediate test model, called SCOTEM (State COllaboration TEst Model) to generate test paths.	A state-based approach to integration testing based on UML models by Shaukat Ali, Lionel C. Briand, Muhammad Jaffar-ur Rehman, Hajra Asghar, Muhammad Zohaib Z. Iqbal, Aamer Nadeem[33]
State Transition Diagrams	Statistical functional testing using transition coverage	Automated generation of statistical test cases from UML state diagrams by Chevalley, P Thévenod-Fosse [34]

Activity Diagrams		A UML-based approach to system testing by Jean Hartmann, Marlon Vieira, Herbert Foster and Axel Ruder[35]
Activity Diagrams	Activity Path Coverage	A Novel Approach to Generate Test Cases from UML Activity Diagrams by Debasish Kundu and Debasis Samantha.[36]
State Diagrams		Synthesis of System State Models by Monalisa Sarma and Rajib Mall[37]
Activity Diagram		Using Adaptive Agents to Automatically Generate Test Scenerios from the UML Activity Diagram by Dong Xu, Huaizhong Li, Chiou Peng Lam[38]
State Diagrams	executable models for state diagrams that can depict workflow related application functionalities that rely on logical implementations	Model Based Testing – Executable State Diagrams by Annamariale Chandran[39]
Statechart Models	applied extended grammar and path generation algorithms to generate testable state charts	Test Ready UML Statechart Models by PVR Murthy, PC Anitha, M Mahesh, Rajesh Subramanyan.[40]
	mutation analysis	UML in Action: A Two-Layered Interpretation for Testing by Bernhard K. Aichernig, Harald Brandl, Elisabeth Jobstl, Willibald Krenny.[41]
Activity Diagrams	object graphs are derived from activities	UML Activity Diagram-Based Automatic Test Case Generation For Java Programs by Mingsong Chen, Xiaokang Qiu, Wei Xu, Linzhang Wang, Jianhua Zhao And Xuandong Li.[42]
	Aggregate model	Testing UML Designs by Orest Pilskalns, Andrew Knight.[43]
Class Diagrams	object graphs are derived from activities	SMT Based Enumeration of Object Graphs from UML Class Diagram by Kenro Yatake, Toshiaki Aoki.[44]
Activity Diagrams	extract test information from activity diagram in the form of input-output sequence and parameters, constraint conditions and expected object method sequence.	Generating UML Test Cases from UML Activity Diagram Based on Gray Box Method by Wang Linzhang, Yuan Jiesong, Yu Xiaofeng, Hu Jun, Li Xuandong and Zheng Guoliang.[45]
Activity Diagrams		A Proposed Test Case Generation Technique Based on Activity Diagrams by Pakinam N. Boghdady, Nagwa L. Badr, Mohamed Hashem and Mohamed F.Tolba.[46]
		Early Fault Detection with Model-Based Testing by Jonas Boberg.[47]

9. Conclusion And Future Work

There are a lot of approaches which tell how to derive test cases from various UML diagrams. Each approach has both drawbacks, and advantages. But none of them describes the whole process with the stated algorithms. We have listed various approaches based on UML diagrams, and the Use Case based approaches have been described too. As future work, these approaches could be further compared and analyzed for determining the best approach.

References

[1] Suman Kasnia and Deepa Mehta. Validating and Testing Software System using Use Case Based Approach. Published in International Journal of Computational Engineering & Management (IJCEM), Vol. 13, July 2011.

[2] Be Careful With “Use Cases” by Edward V. Berard. The Object Agency, Inc.

[3] D. Wood and J. Reis. Use Case Derived Test Cases. www.ittestpapers.com.

[4] http://www.allaboutrequirements.com/2011/10/test-cases-based-on-use-cases.html August 30, 2012

[5] M. Katara and A. Kervinen. Making Model-Based Testing more Agile: a Use Case Driven Approach. In proc. HVC'06 Proceedings of the 2nd international Haifa verification conference on Hardware and software, verification and testing Pages 219-234 Springer-Verlag Berlin, Heidelberg ©2007

[6] Alan Hartman, Mika Katara2, and Sergey Olvovsky. Choosing a Test Modeling Language: a Survey. In Proceedings of the Haifa Verification Conference 2006, LNCS, Springer.

[7] C. Nebut, F. Fleurey, Yves Le Traon and Jean-Marc Je´ze´quel. Automatic Test Generation: A Use Case Driven Approach. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 32, NO. 3, MARCH 2006

[8] Approaches for testing use cases – www.lsi.us.es/~javierj/approaches.htm

[9] Javier J. Gutiérrez, María J. Escalona, Manuel Mejías, Jesús Torres. Generation of test cases from functional requirements. A survey.

[10] Erik Kamsties, Klaus Pohl, Sacha Reis, Andreas Reuys. Testing Variabilities in Use Case Models. In proc., Software Product-Family Engineering – 5th International Workshop (Siena, Italy, November, 2003), Lecture Notes in Computer Science, LNCS 3014, pp. 6-18, Springer, Heidelberg, 2004.

[11] Kwang Ik Seo, Eun Man Choi. Comparison of Five Black-box Testing Methods for Object-Oriented Software. In Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications (SERA ’06), IEEE.

[12] Arilo C. Dias Neto, Rajesh Subramanyan, Marlon Vieira, Guilherme H. Travassos. A Survey on Model-based Testing Approaches: A Systematic Review. WEASELTech’07, November 5, 2007, ACM.

[13] http://www.softwaretestingmentor.com/test-design-techniques/usecase-testing.php

[14] Hans-Gerhard Gross. Component-Based Software Testing with UML. ISBN 3-540-20864-X Springer Berlin Heidelberg New York.

[15] Marlon Vieira, Johanne Leduc, Bill Hasling, Rajesh Subramanyan, Juergen Kazmeier. Automation of GUI Testing Using a Model-driven Approach. In Proceedings of AST’06, May 23, 2006, ACM.

[16] http://www.ibm.com/developerworks/web/library/wa-aj-testing/

[17] D. Kundu, M. Sarma, D. Samantha. A Novel Approach to System Testing and Reliability Assessment Using Use Case Model. ISEC’08 February 19-22, 2008. ACM.

[18] A.L.L. de Figueiredo, W.L. Andrade, P.D.L. Machado. Generating Interaction Test Cases for Mobile Phone Systems from Use Case Specifications. ACM SIGSOFT Software Engineering notes. Volume 31 Number 6. November 2006.

[19] Lasse Harjumaa and Ilkka Tervonen. Introducing Mitigation Use Cases to Enhance the Scope of Test Cases. In Proceedings of Advances in Information and Computer Security, Lecture Notes in Computer Science Volume 6434, 2010, pp 337-353.

[20] Christian Franz Josef Lange. Assessing and Improving the Quality of Modeling: A Series of Empirical Studies about the UML

[21] Darko Gvozdanović, Saša Dešić, Darko Huljenić. UML SUPPORTED SOFTWARE DESIGN

[22] SANTOSH KUMAR SWAIN, SUBHENDU KUMAR PANI, DURGA PRASAD MOHAPATRA. MODEL BASED OBJECT-ORIENTED SOFTWARE TESTING. In Journal of Theoretical and Applied Information Technology, 2005-2010.

[23] Emanuela G. Cartaxo, Francisco G. O. Neto and Patr´ıcia D. L. Machado, "Test Case Generation by means of UML Sequence Diagrams and Labeled Transition Systems", IEEE 2007.

[24] Ashalatha Nayak, Debasis Samanta: "Automatic Test Data Synthesis using UML Sequence Diagrams", in *Journal of Object Technology*, vol. 09, no. 2, March{April 2010, pp. 75{104,

[25] Li Bao-Lin, Li Zhi-shu, Li Qing, Chen Yan Hong ,” Test Case automate Generation from UML Sequence diagram and OCL Expression”, International Conference on Computational Intelligence and Security 2007, pp 1048-52.

[26] A.V.K. Shanthi and G. Mohan Kumar, "Automated Test Cases Generation from UML Sequence Diagram", in *International Conference on Software and Computer Applications (ICSCA 2012) vol. 41*

[27] *A.V.K. Shanthi, Dr.G.Mohan Kumar, "Automated Test Case From UML Diagram Using Data Mining Approach",* CiiT International Journal of Software Engineering and Technology, Vol3.No3, March 2011.

[28] A.V.K. Shanthi, Dr.G.Mohan Kumar, "Automated Test Cases Generation For Object Oriented Software", Indian Journal of Computer Science and Engineering, Vol:2, issue 4,Sep2011.

[29] UML-Based Statistical Test Case Generation by Matthias Riebisch, Ilka Philippow and Marco Götze in LNCS 2591, Springer-Verlag Berlin Heidelberg 2003

[30] *Ostrand T. J., Balcer M. J. 1988. Category-Partition Method. Communications of the ACM. 676-686.*

[31] *http://www.lsi.us.es/~javierj/investigacion_ficheros/GenerationAbstract.pdf*

[32] Peter Fröhlich and Johannes Link. Automated Test Case Generation from Dynamic Models.

[33] Shaukat Ali, Lionel C. Briand, Muhammad Jaffar-ur Rehman, Hajra Asghar, Muhammad Zohaib Z. Iqbal, Aamer Nadeem. A state-based approach to integration testing based on UML models.

[34] Automated Generation of statistical test cases from UML state diagrams by Chevalley, P in Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International

[35] A UML-based approach to system testing by Jean Hartmann, Marlon Vieira, Herbert Foster and Axel Ruder in Innovations Syst Softw Eng (2005) 1: 12–24 Springer, Verlag 2005

[36] Debasish Kundu and Debasis Samantha, A Novel Approach to Generate Test Cases from UML Activity Diagrams, Journal of Object Technology, June 2009

[37] Monalisa Sarma, Rajib Mall, Synthesis of System State Models, *ACM SIGPLAN Notices*, *Vol. 42 (11), Nov. 2007*

[38] Dong Xu, Huaizhong Li, Chiou Peng Lam, Using Adaptive Agents to Automatically Generate Test Scenerios from the UML Activity Diagram, *IEEE Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05), 2005*

[39] Annamariale Chandran. Model Based Testing – Executable State Diagrams, *Published in STEP-AUTO 2011, AVACorp Technology, Chennai.*

[40] PVR Murthy, PC Anitha, M Mahesh, Rajesh Subramanyan. Test Ready UML Statechart Models, SCESM '06 *Proceedings of the International Workshop on Scenarios and State Machines : Models, Algorithms, and Tools, 2006*

[41] Bernhard K. Aichernig, Harald Brandl, Elisabeth Jobstl, Willibald Krenny. UML in Action: A Two-Layered Interpretation for Testing, *Published in ACM SIGSOFT January 2011 Volume 36 Number 1.*

[42] Mingsong Chen, Xiaokang Qiu, Wei Xu, Linzhang Wang, Jianhua Zhao And Xuandong Li. UML Activity Diagram-Based Automatic Test Case Generation For Java Programs, *The Computer Journal Advance Access published August 25, 2007.*

[43] Orest Pilskalns, Andrew Knight. Testing UML Designs, *Published in Information and Software Technology (49), 2007, Science Direct.*

[44] Kenro Yatake, Toshiaki Aoki. SMT Based Enumeration of Object Graphs from UML Class Diagram, *ACM SIGSOFT Software Engineering Journal, July 2012 Volume 37 Number 4.*

[45] Wang Linzhang, Yuan Jiesong, Yu Xiaofeng, Hu Jun, Li Xuandong and Zheng Guoliang. Generating UML Test Cases from UML Activity Diagram Based on Gray Box Method, *Proceedings of 11th Asia Pacific Software Engineering Conference, IEEE '04.*

[46] Pakinam N. Boghdady, Nagwa L. Badr, Mohamed Hashem and Mohamed F.Tolba. A Proposed Test Case Generation Technique Based on Activity Diagrams, *International Journal of Engineering & Technology IJET-IJENS Vol: 11 No: 03 June 2011*

[47] Jonas Boberg, Early Fault Detection with Model-Based Testing, *Erlang'08, September 27, 2008, Victoria, BC, Canada ACM 2008*