International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 586-596 June 2018-Dec 2018 www.csjournals.com

Test Case Generation Tools-A Review

Rajvir Singh[1], Anita Singhrova[2], Rajesh Bhatia[3]
[12] Deaprtment of CSE, Deenbandhu Chhotu Ram University of Science and Technology, Sonepat, India
[3]Department of CSE, PEC University of Technology, Chandigarh, India

Abstract: In recent years, various researchers have been working on the tools or methods to optimize the testing time. The test case generation tools are found to be helpful to reduce the testing time and cost of resources required during testing phase. The existing literature presents various tools supporting procedural as well as object oriented languages. Test case generation is main step in software testing. These tools are helpful for automation of test case generation. This paper presents a review of test case generation tools. This paper will be helpful for those researchers who are working for the automation of software testing making use of tools. Further, this paper will also be helpful for academic researchers who are working in the area of software test automation.

Keywords: Object-Oriented software testing tools, Software testing, Software testing tools, open source tools.

Introduction

Software testing is the process of executing the program with the motive to reveal faults. The software testing tools for generation of test cases play a vital role in automation of software testing which in turn reduces the testing time and effort as compared to manual testing. Several methods are used for the generation of test cases by various researchers.

Object oriented testing is used these days in most industries that develop software systems. This paper presents test case generation(TCG) tools. Automatic test case generation is act of using system to identify truth table of tests based on set of contexts and outcomes. Automatic Test case generation enables the tester in speeding up testing cycles with reduced effort and cost. It also helps in maintaining test cases and increases the efficiency of software testing techniques[1]. The variation on each context on inputs during testing increases the total permutations of test cases which are to be executed to test the software system. Further, the existing study presented by Mahadik et. al.[1] discusses only fifteen tools.

The rest of the paper is organized in five sections namely: review method, automated test case generation tools, analysis of study, conclusion and future work, and finally references.

Review Method

This section presents the review method followed for the collection of data for different test case generation tools:

Inclusion / Exclusion Criteria

The inclusion / exclusion criteria for the selection of test generation tools are based on the availability in literature. The different digital libraries: IEEE Explore, Springer Link, citeseerX, ACM digital library, Science direct DL, Wiley DL, google scholar have been searched for the test generation tools. The search string combinations used are: ((Test) + (tools) + (software testing tools) + (test case generation tools)). Total 5404 articles were found. The exclusion criteria is applied: (i) on the basis of title 654 (ii) on the basis of field of study and abstract 4580 articles excluded. (iii) 60 tools are selected after further manual exclusion based on the availability of online documentation.

The data related to these sixty tools was extracted and analysed. Some data from the website links as presented in references section were analysed. The next section provides the table of these 60 tools and their corresponding data extracted.

Automated Test Case Generation Tools

The existing literatures have presented the following sixty test generation tools based on the search criteria. The brief discussion on these tools is presented below:

Hartman et. al.[2] proposed a model based TCG tool named as ADGES which used commercial UML modeling framework. The authors advocated that this tool is mature enough. It takes model labeled transition system specifications of UML model as input and creates test cases as output.

Boshernitsan et. al.[3] presented AgitarOne which is successor of Agitar tool and is a commercial tool for generation test cases from java source code. It analyses java classes for different types of input data and creates observations depicting behavior of class. These observations are then transformed to assertions by the developer using JUnit[36] tool.

ALLPAIRS[4] is a web based open source tool which is freely available for its use. It creates minimized executable test cases, taking .txt file as input. It uses boundary value analysis(BVA), equivalent class partitioning black-box methods for TCG.

Barnett et. al.[5] presented AsmL tool which is a finite state machine(FSM) based test cases generation tool.

Lakhotia et. al.[6] proposed Austin tool which is a C language code search based TCG tool and is available as open source. It supports the procedural code analysis.

Romberg et. al.[7] advocated AutoFocus a web based tool created at university of California. This tool produces packet header traces and is free for academic and research purpose.

Koch et. al.[8] proposed AutoLink tool for test generation using SDL specifications at Institute of Telematics, Germany. Autolink is commercial tool which generates complete TTCN test suite.

AutoTest presented by Meyer et. al.[9] uses traces in design for TCG and minimizes them. It works in integration with EiffelStudio IDE and is class based open source project.

C++Test[10] is a unit level TCG tool for C++ programming which uses random, static analysis method for TCG.

Tanno et. al.[11] presented CATG[11] tool developed at university of Illinois by their team of academic researchers. This tool produces test cases using random symbolic execution of java byte code and is available as open source code.

CertifyIT[12] is commercial tool which is a UML model structural coverage based tool. It takes FSM specifications as input and produces pest paths.

Class Visualizer[13] is an analysis to which generates class dependency graph from java byte code and available open source.

CodeProAnalytix[14] class based OO TCG tool used for generating JUnit test cases from java source code. It is product of Google Inc available for commercial use.

CompleteTest[15] is a model based tool developed at Malardalen university. It is structural coverage based tool and is available for academic use.

Conformiq[16] is model based tool which is available for commercial use and is a path coverage based tool.

Nakornburi et. al.[17] advocated a pair-wise test generation tool called as CPTG. This tool generates test cases based of statistical user profile data in the form of .csv file.

Cute and jCute are the tools proposed by Sen et. al.[18][31] and are used to generate test data from C code and java code respectively. These tools are open source and available free for academic and research purpose.

Burnim et. al.[19] presented Crest an open source tool used for TCG for C language programs. It uses dynamic symbolic execution method.

Daikon[20] tool generates test sequences from java source code and is integrated with JUnit[36] testing environment. It is available as open source.

Godefroid et. al.[21] proposed DART a code based object oriented testing tool which takes DART source code as input and gives test cases as output.

DIVERSITY[22] is an open source tool for generating test cases. It generates TTCN-3 based test cases.

Fraser et. al.[23] presented Evosuite tool for executable test cases generation. Evosuite takes java source / byte code as input and creates genetic algorithm based test cases. Evosuite is available as open source software.

Cadar et. al.[24] advocated KLEE tool which takes C bit code as input and creates test cases based on concolic method. It is available as open source software.

Findsbugs[25] is java source code based open source tool. It performs module level static analysis.

FMBT[26] developed by Intel is an open source tool which performs TCG from models written in Python or AAL (a pre-post condition language) and is capable of online and offline testing on Linux platform.

GATel[27] generates test cases from Lustre language written constraints specification programs and is available for academic use.

Ma et. al.[28] proposed GRT tool for test data generation which performs random static and run time analysis. It takes java source / byte code as input. GRT is available freely for academic or research use.

HTG[29] is an academic TCG tool for hybrid systems which takes hybrid automation model and creates C++ test cases.

Jcrasher, a code based tool presented by Csallner et. al.[30] is used for object oriented testing. Java source codes are taken as input and test file are generated as output. This is an open source tool under the Apache license.

JMT[32], a code based testing tool which is used for performance evaluation and modeling. Main features of this tool are Mean value analysis, Simulation, and performance engineering.

Albert et. al.[33] developed jPET, a code based testing tool which uses java source code as input and develop test case for that. It is annotated with JML specifications and uses symbolic execution technique for test cases generation.

JTest[34] is a code based testing tool for Java Language programs made by parasoft. This can be used for static analysis, unit test case generation, regression testing, code review and many more. But this is a commercial tool.

Sakti et. al.[35] proposed JTExpert, an academic tool which uses code based approach for testing java source code. It uses search based technique for test case generations. Main focus of this tool is on branch coverage.

Junit[36] is also a Java testing tool which uses code based approach. It's an open source tool which uses builder pattern for test case generation.

Simons et. al.[37] proposed Jwalk tool which is an academic tool used for Java testing using code based approach. It uses specification bases or random systematic technique for test case generation.

Boyapati et. al.[38] proposed Korat an open source tool for java source code testing which follows code based approach. It uses constraint based techniques for test case generation.

Lurette[39] is an academic tool which uses model based approach and generation test sequences. It mainly focuses on transition based coverage method.

Lutess tool presented by Bousquet et. al.[40] follows constraint based generation method and generates test sequences. It's an academic tool used for lutes sequence generation from environment model.

MISTA[41] is a model based testing tool which is used for testing different language programs such as Java, PHP, C#,C,C++,Python.

Modbat[42] , a model based testing tool which uses random and stochastic search techniques for test cases generation. It is open source tool , free to use but not to modified.

ModelJUnit[43] is an open source tool which takes UML models as input and generates class dependency Graph. It focuses on Path coverage.

MoMut[44] , an academic tool which is used for test case generation from models. It uses UML state machine, timed automata or Model specification transition code as input and search based and fault based techniques for test case generation.

Albert et. al.[45] proposed PET, a code based testing tool which is used for white box testing uses Java source code as input and generates test cases. It's an academic tool which uses symbolic execution for test case generation.

PEX tool presented by Tillmann et. al.[46], a code based testing tool applies dynamic symbolic execution for test case generation. With this tool, parameterized unit test cases can be generated and resulting in high coverage.

PICT [47] is a model based tool used for object oriented testing. It is an open source tool. It is used to develop test cases and test configurations for system programs. PICT is a command line tool and used to gain combinatorial coverage.

PragmaDev [48] is a Model and testing tool. It integrates four different tools such as SDL, TTCN-3, SDL-RT and UML. It is used for symbolic execution and in gaining structural coverage in system program.

Pacheco et. al.[49] proposed Randoop, a code based tool which uses java source code as input and generates unit test case as output. It is an open source tool. It uses feedback directed random test generation techniques.

Pacheco et. al.[50] & Pasareanu et. al.[52] advocated Symbolic path finder (SPF) tool which takes java byte code as input. It is used to generate test cases automatically and check errors in java byte code program. In SPF, symbolic execution with model checking is used to perform this work.

Clarke et. al.[51] proposed STG a model based tool which generates test cases for LTS. This tool takes FSM file as input and produces test paths as output. STG is an academic tool which uses NTIF language.

Prasetya et. al.[53] proposed T3, a code based tool used to test java programs. T3 is an open source tool under GPL Ver. 3. Random dynamic testing is used to generate test cases for java programs.

Tcases[54] is an model based tool used for object oriented testing. Basically, XML files are taken as input and junit test cases are generated as output. It is an open source tool used for black box testing to gain structural coverage and constraints solving.

Muniz et. al.[55] presented TCG, a model based testing tool used for functional and statistical testing. Techniques which are used in this tool are Search based , random Generation, and specification based. It's an open source tool and free to use.

Testcomposer[56] is a object oriented testing tool used for model driven application testing. It goes under the commercial license. It can also be used for remote server procedures testing and local testing.

TestGen4j[57] , a code based tool which is used to test java programs. It takes java programs as input and generated unit test cases as output. This tool basically focuses on boundary value analysis.

TGV[58] is model based test case generation tool which uses LOTOS files as input. This tool permits generation of abstract test cases. It is an academic tool used for LTS.

Herout et. al.[59] proposed UMLTest, a model based tool used for object oriented testing. It focuses on gaining path coverage by taking XML files as Input and generating test paths as output.

VERA [60] is a web based tool used for object oriented testing. This is an academic tool. In VERA, Search based techniques are used for generation of test cases. Main focus of this tool is on Structural coverage.

The below is the summary presented in table 1 and following abbreviations are:

Column Headings: TT-Testing type, LS-Language Support, AL – Availability / License, TTM-Testing Technique / Method.

Testing Types: MB- Model Based, CB- Code based, WB – Web Based.
Language Support type: OO- Object Oriented, P- Procedural, T- Test framework.
Input: JSC- Java Source Code, JBT- Java Byte Code, FBD – Function block diagram, MS – Model Specifications, TF- .txt file, FSM- Finite State Machine, CSC- C Source Code, CLP – Constraint Logic Program.

Table 1: Test generation tools

Sr. No.	**Tool Name**	**TT**	**LS**	**Input**	**Output**	**Institution / Authors**	**Domain**	**AL**	**Language**	**TTM**
1	AGEDIS	MB	OO	MS	Test cases	A. Hartman and K. Nagin, IBM Research labs	Desktop / downloadable	Academic/ Commercial /Research	AGEDIS Modeling/ UML /AML	LTS
2	Agitar	CB	OO	JSC	Junit test cases	M. Boshernitsan, R. Doong, A. Savoia, at Agitator Technologies	Desktop	Commercial	Java	Observation Driven
3	AgitarOne	CB	OO	JSC	Junit test cases	Agitator Technologies	Desktop	Commercial	Java	Random Observation Drived
4	ALLPAIRS	WB	OO	TF	Minimized Executable Test cases	James Bach /Microsoft	Online web link	open source	Python / .txt file	BVA, Equivalence partitioning, pairwise combinations method, Combinatorial, black-box testing
5	AsmL	MB	OO	FSM	Test case sequences	Microsoft	Desktop	Research Institute	AsmL	FSM
6	Austin	CB	P	CSC	test cases		Desktop	Open Source	C	Search Based

7	AutoFocus	WB	T	CLP	Packet header traces	Cristian Estan, University of California, San Diego	Online downloadable	Free for Academic purpose or research purpose / downloadable	Autofocus	Constrained Logic Programming
8	AutoLink	MB	OO	SDL, MSC, TTCN	Complete TTCN test suite	Institute of Telematics, Germany	Downloadable	Commercial	Specification and Description Language(SDL), Message sequence chart(MSC), Tree and Tabular Combined Notation(TTCN)	formal specification based
9	AutoTest	CB	OO	C++ / JSC	editable test cases	used and developed by a number of organizations, including Google, IBM, Red Hat, and many others	Desktop/ Downloadable from GitHub / Linux	open-source project under the GPL	Python	Code coverage/path coverage based
10	C++Test	CB	OO	C++ Source & binary code	Unit tests	Parasoft	Desktop	Commercial	C++	Random Static Analysis
11	CATG	CB	OO	JBC	Test cases	Kaushik Sen, University of Illinois	Desktop	Open Source BSD	Java	Symbolic execution, Random testing
12	CertifyIT	MB	OO	FSM - UML	Test cases	Smart Testing - CertifyIt solution	Windows / offline	Commercial	HTML/XML /Perl/Python / Java	Search based, model checking, Structural Coverage, Test based specifications
13	Class Visualizer	MB	OO	JBC	Class Dependency Graph	NA / free	Desktop /Downloadable	Free	Java	class base
14	CodePro Analytix	CB	OO	JSC	Junit test cases	Google Inc.	Desktop / downloadable	Commercial	Java	Random /White Box
15	CompleteTest (uses UPPAAL model checker)	MB	OO	FBD	Abstract test cases/ executable test cases in C / Java	Malardalen University	Desktop	Academic	C / Java	Model checking, Search based structural coverage
16	Conformiq	MB	OO	Model of SUT	Test scripts for automated testing	Conformiq team	Downloadable	Commercial	Java Script	Path coverage based
17	CPTG	CB	NA	CSV file	Test cases	Sompong Nakornburi et. al, Chulalongkorn University, Thailand	Not available	NA	NA	Pairwise testcase generation
18	CUTE	CB	P	JSC	Test cases	K. Sen and G. Agha	Desktop	Open Source / Academic & Research	C	Concolic / symbolic execution
19	Crest	CB	P	C source code	Test cases	GitHub Team	Desktop	Open Source	C	Dynamic test generation / symbolic execution
20	Daikon	CB	OO	JSC	Junit test cases	Massachusetts Institute of Technology & University of	Desktop	Open Source	Java	Observation Driven

						Washington				
21	DART	CB	OO	DART / C source code	test cases Object oriented		Desktop	Open Source	DART /C	Concolic
22	DIVERSITY	MB	OO	xLIA(eXecutable Language for Interaction & Assemblage)	Test cases in TTCN-3	European Telecommunication Standards Institute)		Open Source	xLIA / TTCN-3	Symbolic Transition based
23	EVOSUITE	CB	OO	JSC / JBC	Junit test cases with reduced test suite	Research Project by Dr. Gordon Fraser and Dr Andrea Arcuri	Desktop	Academic Research & Open Source LGPL liscense	Java	Search Based GA
24	KLEE	CB	P	C bit code	Test cases	C. Cadar, D. Dunbar, D. Engler	Desktop	Open Source	C	Concolic
25	Findsbugs	CB	OO	JBC	Test data	Eclipse	Desktop	Open Source	Java	Static Analysis, Unit / Module Testing
26	FMBT	MB	OO	Models in AAL/Python	Test cases	GitHub Team	Desktop: Online / offline Linux /Ubuntu	Open Source	AAL/Python, AAL/C++, AAL/Java	Search based, structural coverage
27	GATeL	MB	OO	Luster Model	Test Cases	French Nuclear Research Agency (CEA)	Downloadable	Academic	Lustre	Constrained logic programming
28	GRT	CB	OO	JSC / JBC	Junit test cases	Leima, Cheng, Hiroyuki, Johannes, Rudolf	Desktop	Academic Research, Free	Java	Random Static & Run time Analysis
29	HTG	MB	OO	Hybrid automation model / SPICE netlist	Test cases in C++	T. Dang	Offline	Academic	C++	Random generation search based, Transition or functional based data coverage
30	Jcrasher	CB	OO	JSC	Test file	Christoph and Yannis	Desktop	Academic Research, Open Source, Apache Liscence	Java	Robustness / random
31	jCUTE	CB	OO	JSC	Unit Tests	Kaushik Sen, University of Illinois	Desktop	Open Source	Java	Search Based Concolic Symbolic Execution
32	JMT	CB	OO	JSC / JBC	OO Metrics				Java	
33	jPET	CB	OO	JSC / JBC	test cases		Linux Desktop	Closed source / commercial	Java	Symbolic Execution
34	JTest	CB	OO	JSC	Junit test cases	Parasoft	Desktop & Server	Commercial	Java	White Box & Static Analysis
35	JTExpert	CB	OO	JSC	Whole test suite	Abedelilah Sakti, Gilles Pesant, Yann-gael, Gueheneuc	Desktop	Academic Research, Free	Java	Search Based
36	JUnit	CB	OO	JSC	Executable Test cases		Desktop	Open Source	Java	Builder pattern

37	Jwalk	CB	OO	JSC	Test report	University of Sheffleld	Desktop	Academic Research	Java	Specification Based / Random Systematic
38	Korat	CB	OO	JSC	Test traces	C. Boyapati, S. Khurshid, and D. Marinov	Desktop	Open Source	Java	Constraint Based Generation
39	Lurette	MB	OO	Lustre Sequence generated from Environment Model	Test sequences	Team of Verimag Group	Online	Academic	Lutin	Transition based coverage
40	Lutess	MB	OO	Lutess Sequence generated from Environment Model	Test sequences	Team of Verimag Group	Downloadable	Academic	Lustre	Constraint Based Generation
41	MISTA	MB	OO	Model Specification transition code(.xmid file)	Test cases	GitHub Team	Online / offline	Open Source	Java, C, C++, C#, PHP, Python, HTML, and VB	Operational, Transition based,structural coverage, Search Based, random generation
42	Modbat	MB	OO	EFSM	Sequences of method calls	Cyrille Artho, Armin Biere		Open Source / free to use bu not to be modified	EFSM / Scala Based domain specific language	Random & Stochastic based Search, FSM
43	ModelJUnit	MB	OO	UML Model	Class Dependency Graph	GitHub Team	Downloadable	open source	Java	Path coverage based
44	MoMuT	MB	OO	Model Specification transition code, UML state machine, Timed automata	Test cases from models	Austrian Institute of Technology,	Offline	Academic	UML State charts, Action systems, Timed Automata, Assume-Guarantee Contracts	Search Based, Mutation / Fault based test case generation
45	PET	CB	OO	JBC	Test cases	Elvira and Miguel	Desktop	Academic Research, Free	Java	Symbolic execution, white Box
46	PEX / INTELLITEST	CB	OO	C# .NET code	parametrized unit tests	Microsoft Liscensed / closed source	Windows Desktop	Academic / Commercial	.NET C#	Dynamic Symbolic Execution
47	PICT	MB	OO	parameter value choices Test Cases	Test Cases and Test Configurations	Microsoft / GitHub	Desktop	Open source	Perl	Combinatorial Testing
48	PragmaDev	MB	OO	MBT Models in SDL	Test cases in TTCN-3	PRAGMADEV SARL, Paris ,France	Windows, Linux, and Mac	Commercial /Free Version small MBT projects	TTCN-3 / C / C++/SDL	Symbolic execution, Structural Coverage, Random and Stochastic
49	Randoop	CB	OO	JSC	Unit test Suite	MIT CSAIL	Desktop	Open Source	Java	Feedback directed random Unit Test Suite
50	SPF (uses javaPathFinder for execution)	CB	OO	JBC	Test paths	GitHub Team	Desktop	Open Source	Java	Symbolic Execution

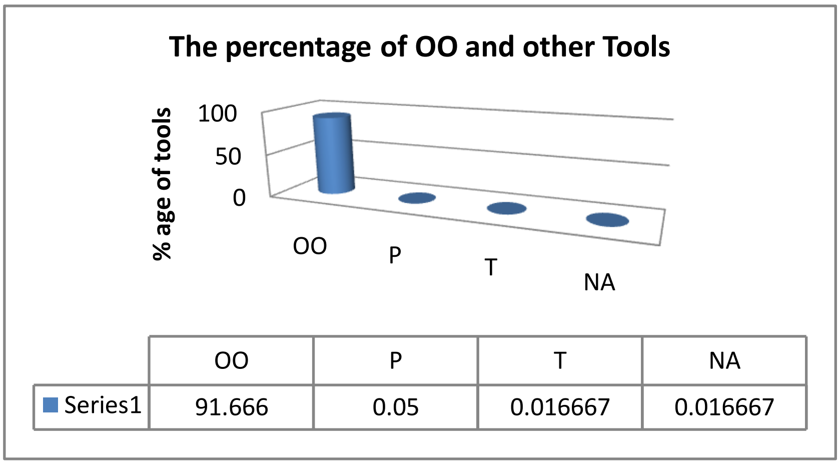
51	STG	MB	OO	LTS / FSM file	Test paths	STG	Downloadable	Academic	NTIF	LTS
52	Symbolic Path Finder	MB	OO	JSC	Test Cases	Microsoft / NASA	Web Application	Open Source Software, used at NASA free for research	Java	Symbolic execution, Model Checking
53	T3	CB	OO	JSC	Test suite	Wisnuprasetya	Desktop	Open Source GPL Ver. 3	Java	Random dynamic testing
54	Tcases	MB	OO	XML file of SUT	Junit test cases	Maven Central Repository:Cornutum Project	Desktop	Open Source	XML / Java	Black box, structural Coverage, Constraints solving
55	TCG - Plugin of LoTuS	MB	OO	mealy machine	Test Cases from both probabilistic and non-probabilistic Models	Microsoft	Windows	Open source / Free	FSM	Search based, random generation, Specification based, pairwise testing
56	TestComposer	MB	OO	FSM model specifications in SDL	Test Paths	Canam Software Labs	Windows	Commercial	SDL	LTS/EFSM
57	TestGen4J	CB	OO	JSC	Unit Test cases	Java Source Inc	Desktop	Open Source	Java	Boundary level
58	TGV - plugin for LoTuS	MB	OO	LOTOS file	Test cases	Verimag	Windows	Academic	LTS-API(LOTOS, SDL,UML)	LTS
59	UMLTest	MB	OO	XML file of SUT (UML class diagram)	Test paths	GitHub Team	Desktop	Open Source	XML	Path coverage based
60	VERA	WB	OO	EFSM	Test cases	Vera's offices ,California	Windows /Linux /Unix	Academic	FSM, XML	Search Based, Structural Coverage

The above table data is helpful for the researchers, for selecting the tool as per testing type, input or output format required, institution or authors, domain like desktop or downloadable or online available, type e.g. academic / open source or commercial, object oriented or procedural, method or technique.

Analysis of study

A. Object Oriented Tools

The percentage of object-oriented(OO) and other tools selected in this paper are shown in below fig 1. This study considered the maximum tools which support OO languages. The most of the tools available for OO testing are java language support tools. The other tools include P-procedural language support tools, T-Test framework, NA- not applicable.

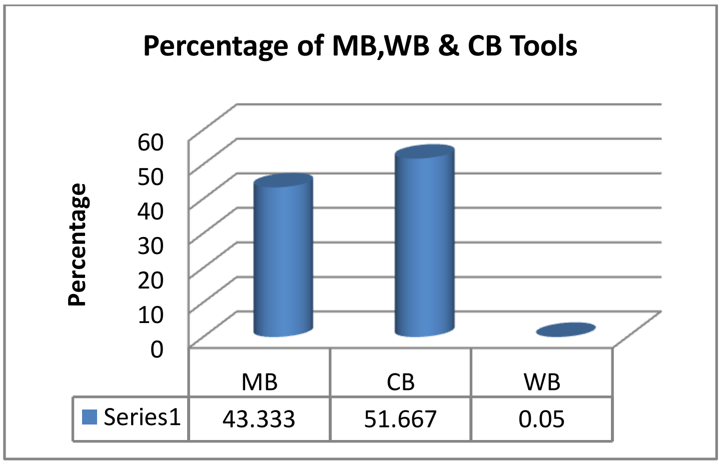


	OO	P	T	NA
■ Series1	91.666	0.05	0.016667	0.016667

Figure 1. The percentage of OO and other tools

B. Test Case Generation Tools based on Testing Type

Based on the type of testing e.g. model based, web based and code based input type, the above tools listed in table 1 are in the percentage as shown in below fig. 2.



	MB	CB	WB
■ Series1	43.333	51.667	0.05

Figure 2. The percentage of OO and other tools

C. Test Case Generation Tools Availablebity

The tools available open source or free for academic / research purposes or commercial are as shown in below fig. 3.

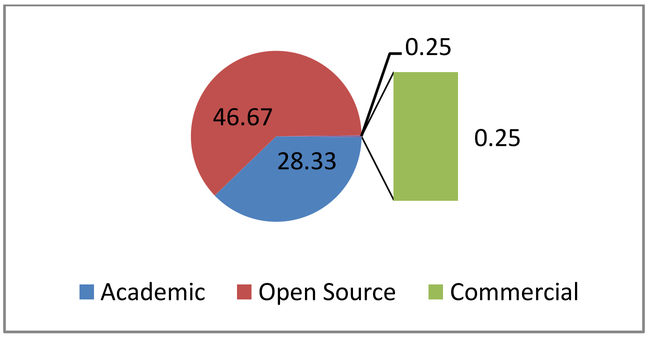


Figure 3. The percentage of OO and other tools

Conclusion and Future Work

In this paper a comprehensive study of various test case generation tools have been presented. In the study the percentage of object oriented test case generation tools indicated that there are a good number of tools available in the field of object oriented testing. The academic researchers are working progressively as the academic tools are 28.33%. The open source tools (46.67%) are available freely to the researchers. Further, the commercial tools are available as 0.25 %, this indicated that commercial tools are not much in use.

References

[1]. P. P. Mahadik and D.M. Thakore, "Survey on automatic test data generation tools and techniques for object oriented code", international journal of innovative research in computer and communication engineering, 2016, 4(1), pp. 357-364.

[2]. A. Hartman, K. Nagin, "The AGEDIS Tools for Model Based Testing", LNCS, 2004, vol 3297. Springer, Berlin, Heidelberg.

[3]. M. Boshernitsan, R. Doong, A. Savoia (2006) "From Daikon to Agitator: lessons and challenges in building a commercial tool for developer testing", In Proc. of ISSTA '06, pp. 169-180. DOI: 10.1145/1146238.1146258

[4]. ALLPAIRS:https://allpairs.soft112.com/

[5]. M. Barnett, et.al.(2004), "Towards a Tool Environment for Model-Based Testing with AsmL", LNCS, vol 2931. Springer, Berlin, Heidelberg.

[6]. K. Lakhotia, M.Harman, H. Gross, "AUSTIN: A tool for Search Based Software Testing for the C Language and itsEvaluation on Deployed Automotive Systems", 2nd International Symposium on Search Based Software Engineering, 2010,pp. 101-110. DOI: 10.1109/SSBSE.2010.21

[7]. J. Romberg, J. Jurjens, G. Wimmel, O. Slotosch and G. Hahn, "AutoFOCUS and the MoDe tool," Third International Conference on Application of Concurrency to System Design, 2003. Proceedings., Guimaraes, Portugal, 2003, pp. 249-250. doi: 10.1109/CSD.2003.1207727

[8]. B. Koch, J. Grabowski, D. Hogrefe and M. Schmitt, "Autolink-a tool for automatic test generation from SDL specifications," Proceedings. 2nd IEEE Workshop on Industrial Strength Formal Specification Techniques, Boca Raton, FL, USA, 1998, pp. 114-125. doi: 10.1109/WIFT.1998.766305

[9]. B. Meyer et al. (2009) "Programs That Test Themselves," Computer, vol. 42, 2009, pp. 46-55. DOI: 10.1109/MC.2009.296

[10]. C++Test: https://www.parasoft.com/products/ctest.

[11]. Tanno, X Zhang, T Hoshino, K Sen ,"TesMa and CATG: automated test generation tools for models of enterprise applications", proceedings of the 37th International Conference on Software Engineering.

[12]. Smartesting CertifyIt. http://www.smartesting.com/en/certifyit/

[13]. ClassVisualizer: https://www.class-visualizer.net/

[14]. Google Inc.," Codeproanalytix user guide", available at: http://developers.google.com/java-dev-tools/codepro/doc/.

[15]. CompleteTest. http://www.completetest.org/about/

[16]. Conformiq: https://www.verifysoft.com/en_conformiq_automatic_test_generation.html

[17]. S. Nakornburi and T. Suwannasart, "A tool for constrained pairwise test case generation using statistical user profile based prioritization," 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), Khon Kaen, 2016, pp. 1-6. doi: 10.1109/JCSSE.2016.7748881

[18]. K. Sen, G. Agha,"CUTE and jCUTE: Concolic Unit Testing and Explicit Path Model-Checking Tools," Computer Aided Verification. CAV 2006, LNCS, vol 4144. Springer, Berlin, Heidelberg.

[19]. J. Burnim and K. Sen (2008) "Heuristics for Scalable Dynamic Test Generation," In Proc. ASE'08, IEEE, pp. 443-446. DOI: 10.1109/ASE.2008.69

[20]. Daikon: https://plse.cs.washington.edu/daikon/

[21]. P. Godefroid, N. Klarlund, and K. Sen. (2005) "DART: directed automated random testing." In Proc. PLDI '05. ACM, pp. 213-223. DOI: 10.1145/1065010.1065036

[22]. Eclipse Formal Modeling Project(DIVERSITY). https://projects.eclipse.org/proposals/eclipse-formalmodeling-project

[23]. G. Fraser and A. Arcuri (2013) "Whole Test Suite Generation," IEEE Trans. on Software Engineering, vol. 39, iss. 2, pp. 276-291. DOI: 10.1109/TSE.2012.14.

[24].C. Cadar, D. Dunbar, D. Engler (2008) "KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs." In Proc. of OSDI'08, pp. 209-224.

[25]. Findsbugs: http://findbugs.cs.umd.edu/eclipse/

[26]. FMBT. https://01.org/fmbt/

[27]. GATeL: https://www.ercim.eu/publication/Ercim_News/enw58/blanc.html

[28]. L. Ma, C. Artho, C. Zhang, H. Sato, J. Gmeiner and R. Ramler," GRT: Program-Analysis-Guided Random Testing", to appear in Int'l Conf. Automated Software Engineering 2015, Nov. (Distinguished Paper Award), 212-223.

[29]. HTG. https://sites.google.com/site/htgtestgenerationtool/home

[30]. C. Csallner and Y. Smaragdakis (2004) "JCrasher: An automatic robustness tester for Java," Software -- Practice ∧ Experience, vol. 34, no. 11, pp. 1025-1050. DOI: 10.1002/spe.602

[31]. Cute: https://cute-test.com/

[32]. JMT: http://jmt.sourceforge.net/Download.html

[33]. E. Albert, I. Cabanas, A. Flores-Montoya, M. Gomez-Zamalloa, and S. Gutierrez," jPET: An Automatic Test-Case Generator for Java", In Proceedings of the 2011 18th Working Conference on Reverse Engineering (WCRE '11). IEEE Computer Society, Washington, DC, USA, 441-442. DOI: https://doi.org/10.1109/WCRE.2011.67

[34]. Parasoft, "Jtest: Java unit testing and code compliance-Parasoft",2007. http://www.parasoft.com /jsp/products/home.jsp?product=Jtest.

[35]. A. Sakti, G. Pesant, Y.-G. Guéhéneuc (2015) "Instance Generator and Problem Representation to Improve Object Oriented Code Coverage," in Software Engineering, IEEE Transactions on, vol.41, no.3, pp.294-313. DOI: 10.1109/TSE.2014.2363479

[36]. JUnit. http://junit.org/junit4/

[37]. Simons, A. J. H.," JWalk: a tool for lazy systematic testing of Java classes by design introspection and user interaction", 2007, Software. Eng., 14 (4), Springer USA (2007), 369-418.

[38]. C. Boyapati, S. Khurshid, and D. Marinov (2002) "Korat: Automated testing based on Java predicates", In Proc. ISSTA'02, pp. 123-133. DOI: 10.1145/566172.566191

[39]. Lurrette. http://www-verimag.imag.fr/Lurette,107.html

[40]. L. du Bousquet and N. Zuanon, "An overview of Lutess a specification-based tool for testing synchronous software," 14th IEEE International Conference on Automated Software Engineering, Cocoa Beach, FL, USA, 1999, pp. 208-215.doi: 10.1109/ASE.1999.802255

[41]. MISTA - Model-Based Testing. http://cs.boisestate.edu/~dxu/research/MBT.html

[42]. Modbat. https://people.kth.se/~artho/modbat/

[43]. ModelJUnit: https://sourceforge.net/projects/modeljunit/

[44]. Momut. https://momut.org/

[45]. E. Albert, M. Gomez-Zamalloa, and G. Puebla, "PET: A Partial Evaluation-based Test Case Generation Tool for Java Byte code", 2010, In Proc. PEPM, ACM, pp. 25-28. DOI: 10.1145/1706356.1706363.

[46]. N. Tillmann and J. de Halleux (2008) "Pex -- White Box Test Generation for .NET," In Prof. TAP'08, Springer, pp. 134-153. DOI: 10.1007/978-3-540-79124-9_10

[47]. PICT: https://github.com/Microsoft/pict

[48]. PragmaDev - Modeling and Testing tools. http://pragmadev.com/

[49]. C. Pacheco and M. D. Ernst," Randoop: Feedback-Directed Random Testing for Java" 2007, In OOP-SLA 2007: Conference on Object Oriented Programming Systems Languages and Applications, ACM.

[50]. C. Pacheco, S. Lahiri, M. Ernst, T. Ball, " Feedback-directed random test generation", Int. Conf. on Software Engineering, ICSE'07, 2007; 75–84, doi:10.1109/ICSE.2007.37.

[51]. D. Clarke, T. Jéron, V. Rusu, and E. Zinovieva. 2001. STG: a tool for generating symbolic test programs and oracles from operational specifications. In Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering (ESEC/FSE-9). ACM, New York, NY, USA, 301-302. DOI: https://doi.org/10.1145/503209.503252

[52]. C. S. Pasareanu et al, "Symbolic PathFinder: integrating symbolic execution with model checking for Java bytecode analysis," 2013, Automated Software Engineering 20:3, pp 391-425. DOI: 10.1007/s10515-013-0122-2.

[53]. W.B. Prasetya,"T3i: A Tool for Generating and Querying Test Suites for Java", 10th Joint Meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE), ACM, 2015.

[54]. Tcases. https://github.com/Cornutum/tcases

[55]. Muniz, L.L., Netto, U.S., Maia, P.H.M.: TCG: a model-based testing tool for functional and statistical testing. In: ICEIS 2015-Proceedings of the 17th International Conference on Enterprise Information Systems, pp. 404–411 (2015)

[56]. Testcomposer:http://www.canamsoftware.com/Products/CAGenSolutions/TestComposer%E2%84%A2/ Overview.aspx

[57]. TestGen4j: https://sourceforge.net/projects/spike-test-gen/

[58]. LoTuS: https://www.ibm.com/developerworks/topics/lotus%20notes%20plugins

[59]. P. Herout and P. Brada, "UML-Test Application for Automated Validation of Students' UML Class Diagram," 2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET), Dallas, TX, 2016, pp. 222-226. doi: 10.1109/CSEET.2016.33

[60].Vera. http://www.spacios.eu/index.php/spacios-tool/