



MODEL BASED TESTING AND UML DIAGRAMS: A LITERATURE REVIEW

Mettu Sumender Roy

Assistant Professor,

Department of C.S.E.,

Vignan's Institute of Information Technology, Visakhapatnam, A.P., India.

Dr. G. Samuel Vara Prasada Raju

Professor in Computer Science,

School of Distance Education, Andhra University, Visakhapatnam, A.P. India.

Abstract

Model based testing is an abnormal state testing method which utilizes abnormal state plan documentations. It has a few points of interest over code based experiment plan. Testing in light of configuration models has the favorable position that the experiments stay legitimate notwithstanding when the code changes a tiny bit. So plan models can be utilized as a premise for experiment era which results in decreasing the expense of testing. The way toward creating test cases from configuration will find issues ahead of schedule in the advancement procedure and along these lines spare time and assets amid improvement of the framework. This paper gives a brief presentation, and general definitions to acquaint MBT, and presents related writing.

Keywords— Code Based Testing, Specification Based Testing, Model Based Testing, System Under Test.

MODEL BASED TESTING

Programming testing is the way toward accepting and confirming that a PC application meets the necessities that guided its outline and advancement, fills in according to desires, can be actualized with the qualities expressed, and fulfills the objectives of partners. By and large, programming testing takes up thirty to fifty percent of the product advancement spending plan, assets and time. Programming disappointments cost organizations beyond a reasonable doubt and customers free a lot of cash. The most noticeably bad programming disappointments have officially harmed notoriety, have negatively affected financials, and made anxiety clients and highlights the proceeding with issues confronted by the saving money and monetary parts. Programming disappointment happens just when there is deficient trying accomplished for the framework. In August 2011, Honda needed to review 2 million autos because of a product issue. The issue needed to do with the modules of transmission control for some of its models. Among the models that were reviewed were the 2001 and 2002 Accord, 2001 to 2003 Civic, 2003 CRV, 2003 Pilot and 2003 Acura 3.2 TL.



General Motors likewise reviewed around 50,000 units of the Cadillac SRX hybrid SUVs in June 2011 in view of a product disappointment that would not let the working of airbags for travelers sitting on the privilege in the back seat in case of a mishap.

Huge speculation must be made on testing as far as time and exertion. Associations ought not quantify testing constructing absolutely with respect to cost and time yet ought to take a gander at the quality it can bring. As indicated by a study it is abridged as: Software is composed by people, people commit errors, and when they commit an error while coding, the product is said to contain an issue or surrender or bug. In case the issue is "executed", a failure may happen. Testing incorporates recognizing insufficiencies before programming gets the chance to be operational. Programming testing constantly remains a psyche boggling, dreary and a slip-up slanted method. In like manner present day mechanical assemblies and advances are ceaselessly endeavoring to organize the relentlessly creating size, versatile quality, convenience and heterogeneity of programming structures. The ability to pass on astounding programming under tight timetables and forceful weight has turned into a critical perspective for programming associations. Spending a lot on superfluous testing can perpetually bring about late conveyance and squanders profitable assets. It is essential to recall, if programming testing can take an organization to its business statures it is the same testing (if not directed appropriately) that can destroy its notoriety for being well as the whole organization

Initially the testing procedure depended on manual work. Manual testing is a procedure that is done to discover the deformities in programming. In this strategy the analyzer assumes the lead part and the end client checks every one of the components of the application. It is an essential sort of testing that discovers bugs in the product. It is a preparatory testing that should be completed preceding computerizing the experiments. For manual testing it is not important to know about any testing apparatuses. Be that as it may, as indicated by programming testing basics complete computerization is impractical. Mechanization in testing has emerged because of the confinements of manual testing. Running the experiments more than once is a difficult procedure in manual testing where as it is simple in computerized testing. Manual testing is not fit for continually changing situations where as computerization in testing is exceptionally helpful when the code as often as possible changes. The part of analyzer is more unmistakable in manual testing and less in mechanized testing. Manual testing is slower in nature while computerization runs test cases altogether quicker than HR.

A propelled approach utilized now days as a part of testing is model-based testing (MBT) where test cases are produced from prior models of the framework under test. The model under test need not be a formal particular of the framework and can only be a representation of a few parts of the necessities to be tried. A



model is a depiction of a framework's conduct. Conduct can be portrayed regarding info successions, activities, conditions, yield, and stream of information from contribution to yield. It ought to be for all intents and purposes reasonable and can be reusable; shareable must have exact portrayal of the framework under test. Model based testing is a trying procedure where run time conduct of programming under test is checked against expectations made by a formal particular or model. In different means, it depicts how framework carries on in light of an activity controlled by a model. Model-based testing is a variation of testing that depends on unequivocal conduct of models that encode the planned conduct of a framework and perhaps the conduct of its surroundings. Sets of info and yield of the model of the execution are deciphered as experiments for this usage: the yield of the model is the normal yield of the framework under test (SUT). As far as model-based testing, the need to accept the model suggests that the model must be less complex than the SUT, or possibly less demanding to check, change and keep up. Something else, the endeavors of approving the model would rise to the endeavors of accepting the SUT. Then again, the model must be adequately exact to serve as a premise for the era of "important" experiments. A model-based testing process must check the included deliberations, and it is likely that oversights in the model imply that these discarded parts can't be tried on the grounds of the model being referred to.

In the course of the most recent two decades the ubiquity of article arranged programming in programming building expanded the development of testing procedures named as model based testing. MBT is a product testing technique is utilized to create test cases totally or somewhat from a conduct model. The experiments produced from this model are theoretical experiments which are not executable. MBT is for the most part accepted as discovery testing on the grounds that the experiments are produced from models rather from the source code. MBT is a strategy for programmed era of experiments utilizing models separated from programming ancient rarities. MBT is a basic approach that depends on experiment era and test outcome assessment. Crucial undertakings in model based testing include gathering the essential data to manufacture a model, producing tests from the model, and assessing the test created from the model. Model is a portrayal of the conduct of programming framework. Conduct is characterized regarding information, yield, activities, conditions, control stream, and information stream in the product framework. There exist numerous models that depict distinctive parts of programming conduct. Some of them are limited state machines, state diagrams, UML, markov chains, petri nets, and choice tables.

Definition of Model Based Testing



Model-based testing is a way to deal with programmed test era utilizing models extricated from programming relics [Dalal et al., 1999].

"Model-based testing is a general term that means a methodology that bases normal testing errands, for example, experiment era and test outcome assessment on a model of the application under test " [El-Far and Whittaker 2001]

"Model-based testing is to utilize express conduct models to encode the proposed conduct and to infer test cases that are utilized for confirming the individual usage." [Pretschner et al., 2005]

"Model-based testing is the advancement of testing relics on the premise of UML models, which give the essential data to creating test cases and test suites, and for checking the last execution of a framework." [Gross 2005] "Model-based testing is the mechanization of the outline of black-box tests." [Utting and Legeard]

"The main thought of model-based testing is to utilize models characterized in programming development to drive the testing procedure, specifically to naturally produce the experiments." [Bertolino]

Advantages of MBT.

1. Improved flaw location. Considers demonstrated that model based testing could reveal the same number of shortcomings or more blames as manual testing could do. Pretschner in his study reported that MBT can recognize bugs six times more than what conventional programming testing systems can do.
2. Reduction in cost and cycle time. The most vital explanation behind utilizing MBT is its productivity and cost funds [Doyle et al., 1997, Dalal et al., 1999]. Via mechanizing the way toward testing MBT can bring down the expense of testing. Making of the model results in utilizing it for any number of times to create test cases which diminishes the individual hours spent on testing. Distinguishing imperfection and altering them at the early stages before advancement begins, is less costly contrasted with identifying and settling deformities after culmination of the framework.
3. Improved test quality and traceability. Nature of the tests is uniform in MBT as the configuration procedure is precise since the model is built from the prerequisites.
4. Defect detection in specification MBT itself goes for revealing the imperfections in the determination since model is produced from necessities and detail. Any errors in the details are followed out utilizing MBT.



5. **Reduced Maintenance costs.** At the point when changes are made to prerequisites, the whole model need not be destroyed rather a few changes are done to it as per the adjustments in the necessity. The adjustment in the model swells through the experiments henceforth decreases the expense of support.
6. **Automation of test generation.** The procedure of Robotization permits reusable tests which minimizes the requirement for manual tests. Robotization decreases the testing time because of the lessened human mediation which builds the item quality and diminishes the venture costs.
7. **Reusability of Assets.** Once a model is made it can be reused for any number of times. Changes to the details permit the model to be reused without tossing it totally which lessens the outline time of a model. Parts of the model or a complete model can be reused.
8. **Achieving better coverage.** Experiment era incorporates accomplishing better scope. Scope is utilized to quantify the nature of the tests. Better representation of prerequisites in the model results in better scope. Better is the scope, better is the model created.

LITERATURE SURVEY ON MBT USING UML DIAGRAMS

1. Pakinam N. Boghdady, Nagwa L. Badr, Mohamed Hashem and Mohamed F.Tolba 2011.

The strategy created by Pakinam et al depends on Basis way testing and Cyclomatic unpredictability. Proposed approach utilizes behavioral models to produce test cases. The proposed model studies the movement outlines as a building hinder in dispatching the robotized calculation of producing experiments. The methodology proposed can create proficient experiments sparing time, exertion and expanding the nature of the experiments produced. Proposed strategy utilizes action graph as information and peruses its substance and stores the information in Activity Dependency table. ADT assembles data with respect to all exercises in the action chart including choices, circles and synchronization, their names, controlling elements, their reliant hubs, info and the normal yield. Utilizing ADT a coordinated diagram is developed and is called as Activity Dependency Graph. To create the action chart the ADT is crossed and each image in the table is spoken to as a hub and move between exercises is spoken to as an edge checking the reliance section. The diagram is crossed utilizing Depth first pursuit to get the test ways. The produced test ways cover all branches, conditions, and premise ways in which circles are secured for zero or one time. Approval of the experiments is done utilizing cyclomatic intricacy system. Cyclomatic movement table contains the exercises included, their names, and their conditions. From the cyclomatic movement table, cyclomatic action chart is produced. From the cyclomatic action chart the quantity of least number of experiments to be secured are ascertained and a lower bound and upper bound qualities are figured. The created test cases apply the branch scope criteria, and the Cyclomatic many-sided quality scope. Proposed technique satisfies full way scope now and again when the



produced test cases surpass the Cyclomatic many-sided quality. Subsequently, the proposed model applies the mixture scope paradigm.

2. Debasish Kundu and Debasis Samanta 2008

The proposed approach produces test cases from Activity chart. It proposes another scope system, movement way scope strategy. Significance of the methodology is

- a. Approach is skilled to distinguish more blames like issue in circles and synchronization shortcomings.
- b. Test cases produced are utilized as a part of recognizing deficiencies in execution.
- c. Approach enhances the outline nature of the framework by discovering flaws and giving early execution.

Proposed technique accumulates important test data from the movement outline, utilizes the data to change over the action chart to action diagram and to produce test cases from the action chart. Points of interest of every action, for example, what are activities epitomized in a movement and what are the inputs, yield parameters of every activity are assembled. Ten sorts of exercises S(start hub), E(flow last/movement last), A(activity), O(object), OS(object state), M(merge), F(fork), J(join), D(decision), C(condition) are considered in this methodology. An arrangement of 12 guidelines are characterized for mapping action chart to action diagram. From the movement diagram test cases are produced utilizing a blend of Depth first pursuit and expansiveness first inquiry calculations. For this reason the diagram is partitioned into simultaneous and no simultaneous sub charts. Profundity first inquiry is connected on non simultaneous sub charts and broadness first pursuit on simultaneous sub diagrams. Three sorts of scope criteria, Basis way scope, basic way scope and Activity way scope are proposed in the methodology. A calculation is proposed to create the experiments. The explanation behind picking Activity outline has been determined in the methodology. "Reasons are credited as takes after: (an) action chart presents ideas at a higher reflection level contrasted with different outlines like arrangement graphs, class graphs and subsequently, movement graph contains less data contrasted with others, (b) nearness of circle and simultaneous exercises in the action graph results in way blast, and for all intents and purposes, it is not attainable to consider all execution ways for testing." The methodology proposed is a model and needs improvement.

3. Santosh Kumar Swain Durga Prasad Mohapatra Rajib Mall 2010

Proposed system is named SATEC (State Activity Test Case era). This strategy utilizes both statechart and action graphs of a framework, changes over them to a halfway shape, produces test cases to accomplish state-



action scope. The transitional representation utilized is state-action chart (SAD). In SAD, the control stream data amid the execution of an utilization case is appeared through a mix of both state moves and exercises. The states are extricated from the statechart graphs and control stream is removed from the movement outline. Pitiful comprises of State-Activity hubs And-Or hubs. State-action hub means the condition of an item amid execution. As well as hubs are further subdivided into 3 sorts and hubs, or hubs as well as join hubs. In as well as hub for and hubs every single active edge are enacted all the while an initiation of all the approaching edges. In or hubs one and only friendly edge is initiated relying on the gatekeeper condition. On the off chance that there are one approaching and different active edges the hub is called or-join hub. Edges interface each hubs in the outline with a Boolean gatekeeper condition related to it. Test scope incorporates state-action scope, move scope and action way scope. In the wake of producing the experiments the methodology assesses the experiments utilizing an issue infusion procedure called transformation examination. Transformation administrators connected in this methodology are change parameter, wrong introductory state, expel capacity call, missing condition, adjust condition administrator, abused watchman condition, missing called capacities, clashing state, missing communication, circle issue, and synchronization fault. The proposed methodology is implied for is implied for group level testing where object associations are tried by considering state-moves of items and the comparing exercises occurring in an utilization case. A model apparatus has been executed in view of the methodology.

4. Raida Elmansouri¹, Houda Hamrouche² and Allaoua Chaoui¹ 2011

In this approach an apparatus is proposed to change UML movement graphs to Communicating Sequential Processes(CSP). CSP is a formal dialect for depicting examples of communication in simultaneous frameworks. The name of the proposed apparatus is named ATOM to change UML action graphs to CSP expressions. The two fundamental errands of ATOM are meta-demonstrating and show change. Meta-demonstrating alludes to displaying formalism ideas at meta-level, utilizing Entity Relationship (ER) formalism or UML Class Diagram formalism stretched out with the capacity to express requirements. Model change in ATOM bolsters chart modifying framework, which utilizes diagram Grammar tenets to outwardly direct the computerized methodology of the change. To change over the action chart to a diagram a novel Meta model and chart punctuation are characterized. Meta model contains seven classes connected by thirty three affiliations. Every affiliation has a characteristic of sort String. It connects a case of the class with a solitary source occurrence of the class definition. Thirty nine creation standards are proposed to produce CSP expressions from movement outline. Generation tenets are made out of two charts on left and right hand sides



(LHS and RHS). Guidelines are contrasted and an info chart called host diagram. On the off chance that a coordinating is found between the LHS of a guideline and a sub-chart in the host diagram, then the standard can be connected and the coordinating sub-chart of the host chart is supplanted by the RHS of the tenet. A modifying framework iteratively applies coordinating guidelines in the language structure to the host chart until no more principles are pertinent.

5. Mingsong Chen · Prabhat Mishra · Dhrubajyoti Kalita 2010

This article presents UML action outline is promising in demonstrating the general framework conduct. This methodology shows naturally creating test cases in light of different model checking procedures. It has three essential commitments.

- Proposing scope driven mapping decides that can naturally make an interpretation of action graph to formal models.
- Proposing a system for programmed property era as per mistake models.
- Various model checking based experiment era methods are characterized to empower proficient experiment era.

The proposed approach diminishes the acceptance exertion by lessening both experiment era time and required number of experiments to accomplish the practical scope. Activity hubs, object hubs and control hubs are considered in this methodology. Control stream and protest stream edges. The proposed method extricates both control and information streams by parsing the UML movement chart. Model interpretation is the way toward mapping control and information streams to the information configuration of Cadence SMV model checker. It comprises of static data extraction and element data extraction. Static data extraction breaks down the structure of a movement graph and after that creates a skeleton of the SMV info. The dynamic data extraction investigates the dynamic conduct of the framework by concentrating on control and information stream examination. To separate the dynamic data an arrangement of guidelines are characterized that indicate the state move for every action hub and the worth changes of the every information. Guideline 1 indicates the interpretation standard for the underlying hub. Standard 2 indicates the interpretation principle for the last hub. Standard 3 characterizes the state changes of a movement. Guideline 4 introduces the state move of the fork hubs. Principle 5 gives the state move of join hubs. Standard 6 demonstrates to control the state change of the move when it is let go. Guideline 7 exhibits the interpretation for quality change of the variables.

In the proposed approach blunder models which are nullification of scope necessities are utilized for experiment era where, a mistake model characterizes an arrangement of blunders for a discretionary



configuration. Every mistake depicted by the model speaks to an arrangement of potential blunders in a configuration. The approval of all blunders can promise the location of all mistakes of the sort secured by the blunder model. Four mistake models action blunder model, move blunder model, key way blunder model and communication mistake models are utilized. The action blunder model can be utilized to check the reachability of every action. The move blunder model can be utilized to check the execution request of the exercises. The key way blunder model is utilized to check the dynamic conduct of the framework. The association blunder model can be utilized to check whether a few exercises can be initiated at the same time. Two experiment era methods are characterized to decrease the general experiment era time. The main strategy utilizes unbounded model checking for experiment era. The second system utilizes SAT-based limited model checking.

6. Baikuntha Narayan Biswal, Pragyan Nanda, Durga Prasad Mohapatra 2008

This procedure utilizes action graph to produce situation based experiments. UML movement graphs are utilized as a part of the methodology since they portray the acknowledgment of the operation in configuration stage furthermore bolster depiction of parallel exercises and synchronization viewpoints required in various exercises impeccably. Test situations are created from action graphs and experiments are produced by dissecting the separate succession and class outline of every situation. In this approach the expense of test model creation is diminished as outline is reused. Name of the proposed methodology is TC-ASEC. Dim box testing is utilized to create the upsides of both black box and white box testing. Action outline is parsed to produce experiment situations which fulfill the way scope criteria which has most noteworthy need among all the scope criteria. Settled fork and join pair is taken care of and afterward whatever remains of the way is produced. The test situation era module is named as TSAD. In the wake of producing the situations arrangement graph and class outline are created for every situation. Arrangement outlines are investigated to discover the connection classifications and after that the class charts are utilized to discover the settings classes. Classification allotment technique is utilized to investigate the utilitarian prerequisites. At that point the experiments are determined by finding critical estimations of environment conditions and parameters. Circles are crossed at most once in this methodology. The proposed method is not completely robotized and an instrument can be created. checking.

7. A.V.K. Shanthi and G. Mohan Kumar 2012

This methodology proposes a heuristic strategy to test the product at the underlying stage to facilitate the product analyzers to test the product. Action outlines are drawn utilizing Rational rose and is spared utilizing augmentation .MDL. A parser is composed to concentrate all conceivable data from the mdl record. A



movement reliance diagram (ADT) is created from the separated data. ADT contains image, movement name, subordinate hubs, include and expected yield. Test ways are created till all states in the table are secured. Each hub in the ADT is doled out a weight with the begin hub's weight is allocated to one, in this way augmenting the quality by one for each next hub navigated. The weights of all test ways are ascertained and hereditary calculation is connected to locate the best experiments. Mechanized device should be produced for this methodology.

8. Monalisha Khandai, Arup Abhinna Acharya, Durga Prasad Mohapatra 2011.

The proposed approach utilizes two UML charts, movement and grouping graphs to show the SUT. Movement Diagram speaks to the succession of action streams and the Sequence Diagram speak to the grouping of messages went between the items. Movement chart has been changed over to action chart and arrangement outline to grouping diagram. Both the charts are joined to a transitional structure called movement grouping diagram. The resultant chart is crossed to create test cases. Scope criteria is connected on movement, succession and action arrangement diagrams and the outcome demonstrates that action grouping chart gives preferable scope over single displaying charts. The proposed approach inspires the disservices of code based testing "certain parts of conduct of a framework are hard to extricate from code yet are effectively acquired from outline models, and experiment era procedure is deferred till the coding is over". To change over the arrangement chart to succession diagram each message ID in the chart is changed to anode and edges speak to the association between the messages. At whatever point a message is passed between two articles in the Sequence Diagram then two hubs are made in the Sequence chart, the hubs are named by message ID of the Sequence Diagram. An edge is doled out between them to demonstrate the reliance among the messages.

A calculation has been proposed to cross the diagram to create test cases is a blend of profundity first hunt and expansiveness first pursuit. Broadness first inquiry is utilized to navigate the simultaneous hubs and profundity first hunt down whatever is left of the diagrams. To change over movement outline to action diagram begin, end, choice, blend, fork, join and action hubs are perceived and their pre edges and post edges are mapped to frame the chart. The chart is navigated utilizing both DFS and BFS to create the experiments. To navigate simultaneous hubs BFS is utilized and to cross whatever is left of the chart DFS is utilized. Presently to produce action grouping outline the Activity Graph is crossed to discover the move starting with one hub then onto the next. At whatever point a move is available two hubs are made in ASG and an edge is relegated between them. The Sequence Graph is crossed to discover the comparing message which is in charge of the move, when the message is found the message is allotted to the edge associating the two hubs. The chart



framed subsequent to covering all moves and messages is the action arrangement outline. The movement grouping diagram is crossed utilizing both DFS and BFS to acquire the experiments.

9. Santosh Kumar Swain, Durga Prasad Mohapatra 2010.

This methodology utilized two charts action and succession graphs to create test cases. The two charts are changed over to diagrams. From the resultant diagram test situations are created. From the test situation the fundamental data for experiment era, for example, strategy action succession, related articles, and limitation conditions are extricated. Focal points of the methodology incorporate

- a. Reduces the quantity of experiments.
- b. Achieves better scope in wording if message-action way scope and class dividing strategy.
- c. Approach covers three deficiencies, message grouping flaws, operation consistency shortcomings from arrangement charts and synchronization issues in movement outline.

The two charts utilized as a part of this methodology are behavioral outlines and both are coordinated into a middle of the road structure called Model stream diagram. Model stream chart created from grouping outline contains an arrangement of hubs and edges where a hub speaks to either a message or control predicate and edge speaks to a move between the hubs. Model stream chart is created for action graph containing hubs and edges where hubs speak to restrictive predicates and edges speak to move between the hubs. The test arrangements are created from Model Flow Graphs where, test successions are an arrangement of hypothetical ways beginning from introduction to end, while taking pre and post conditions into thought. Each produced test arrangement compares to a specific situation of the considered use case. In the following stage test cases are produced from the created successions fulfilling the message-movement way test amplexness criteria. To create test cases all conceivable premise ways are produced from both model stream charts. Each is then gone to for contingent predicates on each of the moves for execution of comparing message and action. For each contingent predicate, class apportioning technique is connected to recognize, to which segment the gatekeeper condition has a place with and the parcels are recorded in the test succession.

10. Ridham Khurana, Anju Saha 2012

This methodology utilized action chart as the model. The methodology plays out a similar investigation of five procedures of test information era utilizing ten cases. The five systems incorporate Test information era utilizing IOAD (info yield unequivocal action chart), Test information era utilizing sub movement graphs, Test information era utilizing condition order tree strategy, Test information era for acknowledgment testing and an



Enhanced experiment era method. Test era utilizing IOAD concentrates on outside collaboration of the framework and overlooks inward preparing it produces test cases from information/yields given and got. From this method all way scope criteria is worked out. In test information era utilizing sub movement charts, a gathering of exercises are considered rather than individual exercises. These gathering are extended to end up a different action graph which is utilized to test fine points of interest in the sub movement outline. This method covers way scope. In test information era utilizing condition grouping tree strategy utilizes all contingent branches to discover which test cases covers all branches. In test information era for acknowledgment testing the movement graph is changed over into association stream outline that shows objects of client's advantage. The diagram produced is crossed for way scope utilizing DFS to create every single legitimate way. In improved experiment era system the action chart is changed over to movement diagram utilizing the arrangement of exercises present in the action graph. Test ways are extricated from the diagram utilizing way scope criteria. Five methods are connected on ten cases and the accompanying examinations are made. Technique 1 and 2 are similar.

- a. Techniques 1 and 3, 2 and 4 are opposite.
- b. Techniques 1 and 4 are similar since both deals with user's perspective.
- c. Technique 1 focuses mainly on concurrent activities.
- d. Technique 2 focuses on conditional classification trees.
- e. Techniques 2 and 5 can be combined to generate efficient test cases.

11. Chen Mingsong, Qiu Xiaokang, and Li Xuandong 2006.

This procedure utilizes petri net like semantics of the action graphs as configuration determinations and creates test cases naturally. The project is executed by the produced test cases to get the execution follows and these follows are contrasted with the movement graph with get a decreased arrangement of experiments which meets the test sufficiency criteria. The methodology is additionally used to check the consistency between project execution follows and the conduct of the movement graph. To assemble the project execution follows program instrumentation is utilized. Its principle thought is to embed a few tests which are one or more test articulations embedded in the first code for recording dynamic data. With instrumentation system hint of genuine execution requests of the part elements of a project are created. For this reason a calculation is proposed. The created follows are checked for consistency with the movement graph. On the off chance that the follow does not coordinate a way in the action graph the test is insignificant. Scope criteria utilized as a part of this methodology is movement scope, move scope and basic way scope. An instrument model has been created for this methodology and is named as AGTCG.



12. Wang Linzhang, Yuan Jiesong, Yu Xiaofeng, Hu Jun, Li Xuandong and Zheng Guoliang 2004

This proposed approach produces test cases from action outline utilizing dim box technique where the configuration is reused to maintain a strategic distance from the expense of test model creation. In this methodology test situations are specifically gotten from the movement outline. From the test situation the contribution/out arrangement, parameters, imperative conditions, expected item strategy grouping are separated for experiment era. The conceivable estimations of all the information/yield parameters are created by applying classification allotment strategy, and test suite could be deliberately produced to discover the irregularity between the usage and the configuration. A model apparatus named UMLTGF has been produced to bolster the proposed approach. The proposed apparatus parses the movement outline and infers the arrangement of test situations to fulfill the essential way scope criteria. At that point, every test situation is handled. The info and yield parameters are removed from the activity grouping. The imperative conditions are extricated from the gatekeeper conditions in every move of the test situation succession. The article technique grouping which speaks to the inner conduct of the product in runtime is removed from action states and comparing objects. Finally classification segment technique is utilized to create legitimate mix of estimations of info and yield parameters to fulfill the condition limitations. So the info succession, expected article technique call succession and expected yield frame an experiment. Ts Generator is the name of the calculation proposed to naturally cross all the movement states and moves of every essential way utilizing DFS to get all test situations from the action chart. Scope criteria utilized as a part of the methodology is choice scope, different condition scope, all edge scope, n-way scope, and all exercises scope.

13. Irum Rauf, Muhammad Zohaib Z. Iqbal, Zafar I. Malik 2009.

This method is utilized to approve the usefulness of composite web administrations and that the web administrations taking part in the structure procedure are good with each other. This methodology contains four modules in particular analyzer module, produce experiment module, instrumentation module and execution module. Dissect module examines the action chart and concentrates data about properties and related interfaces, pre and post conditions including these traits. This rundown is produced in Object requirement dialect. Test ways are created by crossing the action model. Choice scope, different condition scope, all edge scope, n-way scope, and movement scope criteria are utilized to create test ways from the action graph. Test information is created for the main action as whatever remains of the exercises take after the traversal way.



14. Luis FERNANDEZ-SANZ, Sanjay MISRA 2012.

This methodology naturally creates a complete arrangement of practical experiments from UML action graphs supplementing detail of utilization cases. Test cases are organized by danger data. The strategy begins with the action chart which portrays the conduct of an utilization case. The states where information are entered are spoken to with movement states. These action states are utilized to speak to the run of the mill testing values for every information contribution as indicated by conventional limit and comparability dividing procedures. For every edge two variables, likelihood of utilization, significance of capacity are included. Each diverse way which could be followed over the Activity Diagram could be considered as comparable to an utilization case situation. The aggregate number of situations to be broke down has a tendency to be to a great degree high when two certainties are viewed as the quantity of times that a circle is executed inside a general way could make two distinct ways and every info activity of every way can produce independent from anyone else one sub way for every blend of information qualities. Every way is distinguished by the chain of states crossed from introductory state to the last one, including the ones comparing to the particular qualities for contribution to the extended action states.

15. Swगतिका Dalai, Arup Abhinna Acharya, Durga Prasad Mohapatra 2011

THIS METHODOLOGY CONSIDERS TWO OUTLINES SUCCESSION AND MOVEMENT FOR BETTER SCOPE AND DEFICIENCY IDENTIFICATION. THIS METHODOLOGY CREATES TEST CASES FOR SIMULTANEOUS FRAMEWORKS UTILIZING COMBINATIONAL UML OUTLINES. A MIDDLE MODEL TO BE SPECIFIC ARRANGEMENT MOVEMENT CHART IS DEVELOPED FROM THE TWO GRAPHS AND IS CROSSED TO CREATE TEST CASES WHICH CAN MINIMIZE EXPERIMENT BLAST. FROM THE SUCCESSION OUTLINE ARRANGEMENT TABLE IS PRODUCED INCLUDING SOURCE OBJECT, DESTINATION OBJECT, MESSAGE ID, AND MESSAGE CONTENT. SUCCESSION MOVEMENT CHART CONTAINS HUBS FROM ACTION GRAPH WHICH ARE EITHER QUESTION HUBS OR ACTION HUBS. AT THE POINT WHEN A MOVE IS AVAILABLE AN EDGE IS INCLUDED THE GROUPING ACTION DIAGRAM. SUBSEQUENT TO DEVELOPING THE DIAGRAM DFS AND BFS ARE CONNECTED. AT THE POINT WHEN FORK IS EXPERIENCED BFS IS UTILIZED AND FOR WHATEVER IS LEFT OF THE CHART DFS IS CONNECTED. WAY SCOPE CRITERIA ARE UTILIZED TO CREATE TEST CASES.

CONCLUSION

This paper proposes an outline of model in Model Based Testing. This paper abridges a portion of the principle existing methodologies in writing for mechanized testing of UML charts utilizing Model Based Testing.



References

- [1] 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 and Technology IJET-IJENS Vol: 11 No: 03
- [2] Debasish Kundu and Debasish Samanta, "A Novel Approach to Generate Test Cases from UML Activity Diagrams", Journal of Object Technology, Vol: 8 No: 3, May/June 2009.
- [3] Santosh Kumar Swain, Durga Prasad Mohapatra and Rajit Mall, "Test Case Generation Based on State and Activity Models", Journal of Object Technology Published by ETH Zurich, Chair of Software Engineering, © JOT 2010 Online at <http://www.jot.fm>
- [4] Raida Elmansouri, Houda and Allaoua Chaoui, "From UML Activity Diagrams to CSP Expressions: A Graph Transformation Approach using Atom3 Tool", IJCSI International Journal of Computer Science Issues, Vol: 8 Issue 2, March 2011, ISSN (Online): 1694-0814 www.IJCSI.org
- [5] Mingsong Chen – Prabhat Mishra – Dhrubajyoti Kalita, "Efficient test case generation for validation of UML activity diagrams", 18 June 2010 © Springer Science+Business Media, LLC 2010, 105-13-DOI 10.1007/s10617-010-9052-4
- [6] Baikuntha Narayan Biswal, Pragyan Nanda and Durga Prasad Mohapatra, "A Novel Approach for Scenario-Based Test Case Generation", 978-0-7695-3513-5/08 \$25.00 © 2008 IEEE, DOI 10.1109/ICIT.2008.43
- [7] A.V.K. Shanti and G. Mohan Kumar, "A Heuristic Technique or Automated Test Cases Generation from UML Activity Diagram", Journal of Computer Science and Applications, ISSN 2231-1270 Volume 4, Number 2 (2012), pp. 75-86
- [8] Monalisha Khandaj, Arup Abhinna Acharya, Durga Prasad Mohapatra, "Test Case Generation for Concurrent System using UML Combinational Diagram", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2(3), 2011, 1172-1181
- [9] Santosh Kumar Swain and Durga Prasad Mohapatra, "Test Case Generation from Behavioral UML Models", International Journal of Computer Applications (0975-8887) Volume 6-No.8, September 2010
- [10] Ridham Khurana and Anuj Saha "Empirical Comparison of Test Data Generation Techniques using Activity Diagrams", International Journal of Computer Applications (0975-8887) Volume 51-No.7, August 2012
- [11] Chen Mingsong, Qiu Xiaokang, and Li Xuandong, "Automatic Test Case Generation for UML Activity Diagram", AST '06, May 23, 2006, Shanghai, China. Copyright 2006 ACM 1-59593-085-X/06/0005
- [12] I. Wang, J. Yuan, X. Yu, J. Hu, X. Li, and G. Zheng. Generating test cases from UML Activity diagram based on gray-box method. In 11th Asia-Pacific Software Engineering Conference (APSEC '04), pages 28-291, Busan, Korea, 2004.
- [13] Irum Rauf, Muhammad Zohaib Z. Iqbal and RZafar I. Malik, "Model Based Testing of Web Service Composition Using UML Profile", 2nd Workshop on Model-based Testing in Practice (Mo TiP 2009), session 2, paper3.
- [14] Luis Fernandez-Sanz, Sanjay Misra, "Practical Application of UML Activity Diagrams for the generation of Test Cases", Proceedings of the Romanian Academy, Series A, Volume 13, Number 3/2012, pp. 251-260.
- [15] Swagatika Dalai, Arup Abhinna Acharya and Durga Prasad Mohapatra, "Test Case Generation For Concurrent Object-Oriented Systems Using Combinational UML Models", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 5, 2011
- [16] Dalal. S. Jain, A. Karunanithi, N. Leaton, J. Lott, C. Patton, G. and Horowitz, B. 1999, "Model-Based Testing in Practice". In Proceedings of the 21st International Conference on Software Engineering, California, US, May 16-22 1999 IEEE/ACM, 285-294.
- [17] Pretschner. A. Prenninger, W. Wagner, S. Kuhnel, C. Baumgartner, M. Sostawa, B. Z'olch, R. and Stauner. T. (2005), "One Evaluation of Model-based Testing and Its Automation". In Proceedings of the 27th International Conference on Software Engineering, St. Louis, MO, Pages: 392-401, ISBN: 1-59593-963-2. ACM New York.
- [18] Utting M. Pretschner, A., and Legeard, B. (2006), "A taxonomy of Model-based testing". ISSN: 1170-487X, The University of Waikato, New Zealand.
- [19] Ibranhim K. El-far and James A. Whittaker, "Model-based Software Testing" in John J. Maarciniak (Ed), Encyclopedia of Software Engineering (2 volume set), 2nd Edition, Wiley, Dec 2001.
- [20] Hans-Gerhard Gross, Marc Bom, Ina Schieferdecker, Pedro Santos, "Model-Driven Development and Testing – A Case Study". This work has been partially supported by the BMBF project MDTs on Model-Driven Development of Telecommunication Systems, 2005.
- [21] Antonia Bertolino, "Software Testing Research: Achievements, Challenges, Dreams," in Proc. Future of Software Engineering (FOSE 2007), co-located with 29th ICSE 2007, Minneapolis, Minnesota, USA, 23-25 May 2007. IEEE Computer Society Press, 2007, pp. 85-103.
