ISSN 2349-0292
Impact Factor 2.365

Global Journal of Advanced Engineering Technologies and Sciences

AUTOMATED AUTHENTICATION OF USE CASE DIAGRAM USING UMLSECCHECK 3.0 ON XMI DATA

Ananya P*, Dr. Prasad Babu B.R, Dr. Chandramouli S
[*]Department of CSE, East Point College of Engineering & Technology Bangalore, Karnataka, India
Project funded by the Government of Karnataka under VGST Scheme (2015-2016)

Abstract

A Threat in software may cause the failure of system which results in hacking of system. Such threat is caused by improper coding or fault representation of the software. Threat in the coding is well defined and there are many tools to detect the threats but there are only few documents to detect the threat in the UML diagrams. There is need for developers to come up with new tools and technologies which enable them to detect defects in the designs.

In this paper we discusses about a tool called Umlseccheck3.0 tool which is used to detect the threats and to provide security in UML Diagrams. This tool takes the input of XMI data from Argo- UML tool, then it is equated with the others rules which is described for the UML diagrams. The output of this specifies is there any disobedience of the rules which are described for it.

Keywords: Parser, schema, rule file UML models, XMI data.

Introduction

Still just around 4% of programming frameworks by and by are fabricated utilizing displaying procedures or some likeness thereof (the vast majority of them utilizing UML). There should be a persuading increased the value of the utilization of model-based advancement methods before it will be generally received in industry . We will probably give such included quality by creating device support for the investigation of UML models against framework necessities which can be detailed at the level of the framework demonstrate, and which can't be physically checked in a solid and productive path, (for example, security prerequisites). Here, we portray an UML confirmation system supporting the development of mechanized prerequisites examination apparatuses for UML graphs.

Ordinarily, UML models checked against security properties are express models of the framework plan, while in Model-Based Testing (MBT) we depict the normal conduct of an application, considered along these lines to be a black box. With the present cutting edge, on one hand it is workable for a framework architect to plan an origination model explained with security properties that can be confirmed utilizing robotized hypothesis provers and model-checking, for instance utilizing the UMLsec approach. Modeling strategies are utilized as a part of fewer sums in creating programming framework (the greater part of them utilizing UML). Model-based improvement methods ought to be tried before utilizing as a part of industry. Our point is to execute a product examination instrument to test UML models which can't be physically checked in a solid and proficient route, (for example, security necessities). Here, we outline a system for assessing UML models which gives robotized investigation instruments. As a rule the UML charts (model) tried contrary to security measures (properties) are exact graphs (model) of the composed framework. In model based test unsurprising deeds (conduct) of the applications seen as a black box. By utilizing current situation with the workmanship the designer can plan a model reasonably connected with security measures (properties) and can be tried by utilizing model based system.

Methodology

The approach the instrument is appeared in Fig.1 and comprises of three noteworthy useful modules to be specific, Parser, Rule approval and Rule Engine.

UML XMI Data: This information is gotten by changing over UML charts [14] into XMI document design (.xmi augmentation) utilizing Argo-UML apparatus.

Parser: Parser will take the information XMI information and produces

ISSN 2349-0292
Impact Factor 2.365

Test Engine: Security principle sets are composed in view of the application, which is given as a data for standard motor.

Test Validation: At this stage the pattern produced from the parser and principles from tenet motor are considered as information to the tenet approval. The result is tenet infringement/ no infringement got by contrasting pattern and lead sets. This apparatus distinguishes plan defects taking into account any of one: the use case outline, component chart. Future improvements are made arrangements for joining other charts as data.

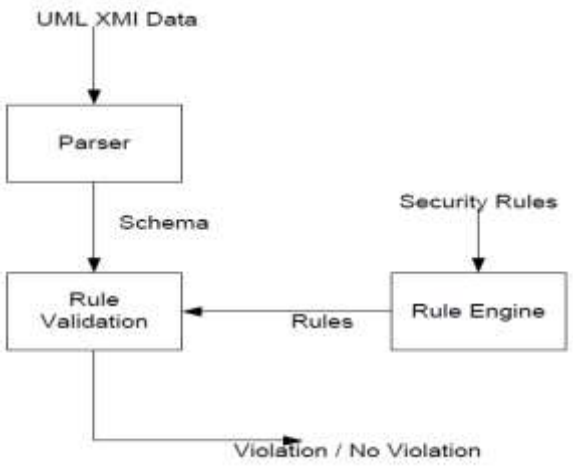

Fig 1: Framework of the UMLSecCheck 3.0 Tool

Use case expert: In this module the use case analyzer distinguishes the use case graph which contains actor name and dependencies. This module likewise distinguishes the quantity of properties and contains quality qualities like number of dependencies which is proclaimed in the use case outline

Test1: is intended for use case outlines which distinguish the actor name and number of dependencies. The test fails if actor name or number of dependencies does not match with the requirement. If the requirement matches then test will pass..

The front end of the created instrument is appeared in Figure.2. It has three screens to be specific, CONFIGURATION, VALIDATION and LOG. Info screen permits picking info UML-XMI Data File and Rule record. When you tap on VALIDATE catch, you can see yield on LOG screen

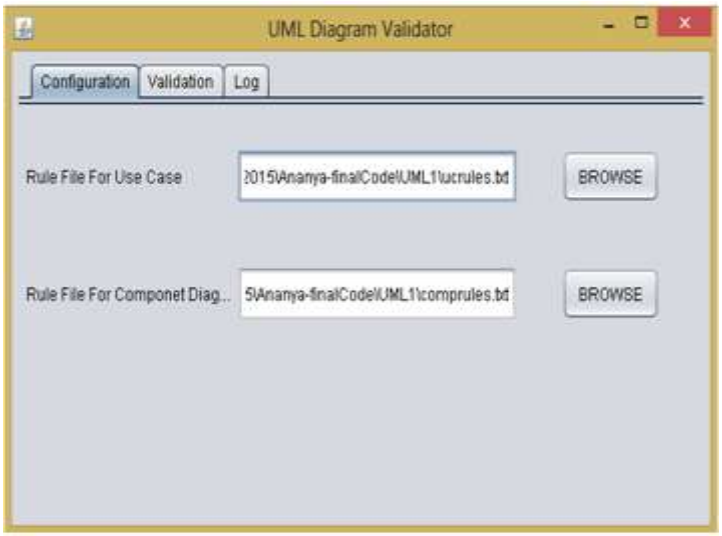

Fig 2: Front end of Tool

ISSN 2349-0292
Impact Factor 2.365

Experimental Results

The instrument was utilized to distinguish vulnerabilities in all the two charts: use case diagram and Component diagram the experimental result is described below.

CASE1: Use case diagram at its most straightforward is a representation of a client's collaboration with the framework that demonstrates the relationship between the client and the distinctive use cases in which the client is included. A use case chart can recognize the diverse sorts of clients of a framework and the distinctive use cases and will regularly be joined by different sorts of graphs too. If the actors name and number of dependencies does not match then the rule will be violated. Hence the actor name and number of dependencies should make to match as the requirements.

Fig 3: An example for use case diagram

TEST CASES: The exploratory results are given for distinctive info UML outlines as appeared in beneath tables

Table1: Test case for use case diagram.

Test case ID	1
Description	Use Case diagram to be tested(Fig.3)
Input	XMI Data Obtained for the input Use case diagram (Fig.3)
Expected Output	Test 1 is violated because actor name and number of dependencies does not matches.
Remarks	Successful

Conclusion

In this paper the activity graphs to meet the needs of a suitable modeling element for use case behavior. The refinement in particular supports a proper coupling of activity graphs and class models. Granularity and symbiology of the approach allow for a seamless, traceable transition of use cases (actually their related activity graphs) to domain classes thus providing the basis not only for consistency and completeness checks but also for the verification of the domain class model against the use case model (comprising use cases and activity graphs). The validation of the use case model and parts of the domain class model is supported as well.

References

1. Gargantini and C. Heitmeyer, "Using model checking to generate tests from requirements specifications," SIGSOFT Softw. Eng. Notes, vol. 24, no. 6, pp. 146–162, 1999.
2. P. E. Ammann, P. E. Black, and W. Majurski, "Using model checking to generate tests from specifications," Formal Engineering Methods, International Conference on, p. 46, 1998.
3. M. B. Dwyer, G. S. Avrunin, and J. C. Corbett, "Patterns in property specifications for finite-state verification," in ICSE'99, 21st international conference on Software engineering, LA, California, United States, 1999, pp. 411–420.
4. C. Jard and T. Jéron, "Tgv: theory, principles and algorithms: A tool for the automatic synthesis of conformance test cases for non-deterministic reactive systems," Int. J. Softw. Tools Technol. Transf., vol. 7, no. 4, pp. 297–315, 2005.
5. L. Frantzen, J. Tretmans, and T. Willemse, "Test generation based on symbolic specifications," in FATES 2004, Formal Approaches to Software Testing, ser. LNCS, J. Grabowski and B. Nielsen, Eds., vol. 3395. Springer, 2005, pp. 1–15.
6. D. Clarke, T. Jéron, V. Rusu, and E. Zinovieva, "STG: A symbolic test generation tool," in TACAS'02, Tools and Algorithms for the Construction and Analysis of Systems, ser. LNCS, vol. 2280. Springer, 2002, pp. 151–173.
7. G. J. Tretmans and H. Brinksma, "TorX: Automated modelbased testing," in 1st Europ. Conf. on Model-Driven Software Engineering, Nuremberg, Germany, Dec. 2003, pp. 31–43.
8. C. Bigot, A. Faivre, J.-P. Gallois, A. Lapitre, D. Lugato, J.-Y. Pierron, and N. Rapin, "Automatic test generation with AGATHA," in TACAS 2003, Tools and Algorithms for the Construction and Analysis of Systems, 9th International Conference, ser. LNCS, H. Garavel and J. Hatcliff, Eds., vol. 2619. Springer, 2003, pp. 591–596.
9. A. Bertolino, E. Marchetti, and H. Muccini, "Introducing a reasonably complete and coherent approach for model-based testing," Electron. Notes Theor. Comput. Sci., vol. 116, pp. 85–97, Jan. 2005.
10. F. Basanieri, A. Bertolino, and E. Marchetti, "The Cow_Suite approach to planning and deriving test suites in UML projects," in UML'02, 5-th int. conf. on the UML language, ser. LNCS, vol. 2460, London, UK, 2002, pp. 383–397.
11. M. Felderer, R. Breu, J. Chimiak-Opoka, M. Breu, and F. Schupp, "Concepts for Model-based Requirements Testing of Service Oriented Systems," in Proceedings of the IASTED International Conference, vol. 642, 2009, p. 018.
12. B. Chetali, "Security testing and formal methods for high levels certification of smart cards," in Proceedings of the 3rd International Conference on Tests and Proofs, ser. TAP '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02949-3_1
13. http://www.wikipedia.org/wiki/Unified_Modelling_Language.html.

Authors Profile:

Ms. Ananya P is M.Tech Scholar in Computer Science and Engineering at East Point College of Engineering and Technology, VTU. She attended and presented papers in National and International conferences in various colleges. Her research areas are Testing, Sensor Networks, IOT and Cloud Computing, Big data..

Dr. Prasad Babu is working as Professor and Head, Dept. of Computer Science and Engineering at East Point College of Engineering and Technology. His research areas are Adhoc networks, Mobile Communication, and Software Engineering. He published more than 50 papers in various International Journals. Presently he is guiding for PhD Scholars in Visvesvaraya Technological University (VTU) and Jawaharlal Nehru Technological University (JNTU) India.

Dr Chandramouli H is working as Professor, R&D - Dept. of Computer Science and Engineering at East Point College of Engineering and Technology. His research areas are Wireless Sensor Networks, Mobile Adhoc Networks, IOT and Cloud Computing. He published more than 10 papers in various International Journals. Presently he is guiding for PhD Scholars in Visvesvaraya TechnologicalUniversity(VTU)India