2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

UML Generated Test Case Mining Using ISA

Lilly Raamesh [1], G.V. Uma [2]

[1] Research Scholar, Anna University Chennai, Chennai – 600 025.

[2] Asst. Professor / CSE, Anna University Chennai, Chennai-600 025.

Abstract. Our paper focuses on the generation of optimal test cases from Unified Modeling Language (UML) diagrams using Intelligent Software Agents (ISA) for highly reliable systems. Knowledge Discovery in Databases (KDD) is a process of finding new, interesting, previously unknown, potentially useful, and ultimately understandable patterns from very large volumes of data. The ultimate goal of data mining is in revealing pattern that is easy to perceive, interpret, and manipulate. Software testing is the most common practice of software quality assurance which needs test cases. In order to improve the efficiency of software testing we go for mining of test cases. One of the most important and challenging problems in data mining is the definition of prior knowledge. This can be originated from the process or the domain. Prior knowledge is helpful for selecting suitable data and mining techniques, pruning the space of hypothesis, representing the output in comprehensible way, and improving the overall method.

Keywords: database, data mining, clustering, association rule, unified modeling language, test cases, and intelligent software agents.

1. Introduction

Testing is a process finding defects. Even when software is developed using a rigorous discipline, it will contain a significant number of bugs. So, testing is needed to guarantee the software quality and reliability before release.

Testing of system behavior can be categorized into

- White box testing or structural testing or glass box testing

In this approach the actual implementation is used to generate the tests. While test generation using the white-box approach is not common, evaluation of test effectiveness often requires use of structural information.

- Black box testing or functional testing

When test generation is done by only considering the input/output description of the software, nothing about the implementation of the software is assumed to be known. This is the most common form of testing.

Data mining algorithms can be applied at different levels of abstraction and help the user discover more meaningful patterns. Data mining will create patterns from the existing database. Using well-established data mining techniques, practitioners and researchers can explore the potential of this valuable data in order to manage their project and to produce higher quality software systems that are delivered on time and within budget.

2. Related Work

Dae-Kyoo Kim and Jon Whittle used a prototype tool called Role Based Modeling Language Pattern Instantiator (RBML-PI) that generates application specific UML class diagrams and sequence diagrams from pattern specifications described in the RBML, a pattern specification language defining a domain specific sub-language of the UML. Domain pattern captures concepts common across an application domain. However, use of these patterns in specific context requires customization of the pattern. [3]

As an expression pattern language based on UML, RBML has the advantage that it does not require the learning of the new language and it can be implemented in existing UML base CASE tools.

RBML-PI uses IBM Rational Rose as base tool. We can use the drawing features provided by Rose to build an RBML pattern specification. Because the RBML is based on UML syntax, application developers can then use the RBML specification to generate UML models in RBML-PI.

Dymek D and Kotulski L proposed a method of supporting the test data generation using the information from different kinds of UML diagrams. This method is based on the UML Vertical Relations (VR) concept which is the extension of the standard UML with VR which define in the formal the relation between elements from different kinds of diagram. Using these VR we are able to use UML as a base for test data generation for different kinds of test (module-based and requirement-based). Such an approach gives us a unique possibility to choose the testing method driven by actual needs. [7]

Last decade has witnessed a very slow but steady advancement made to the testing of object–oriented systems. One significant approach is the generation of test cases from UML models. The main advantage with this approach is that it can address the challenges posed by the object – oriented paradigms. More over, test cases can be generated early in the development process and thus it helps in finding out many problems in design if any even before the program is implemented.

But none of this uses mining approach. In this paper we propose a mining approach to have better knowledge about the test cases and about the full system. Our approach consists of transforming an UML sequence diagram into a graph called the sequence diagram graph (SDG) and the SDG is then traversed to generate the test cases. The test cases thus generated are mined by trained agents.

3. UML

The Unified Modeling Language (UML) is a collection of languages for specifying, visualizing, constructing, documenting the artifacts of the software systems.

4. Generation of Test Cases from UML Models

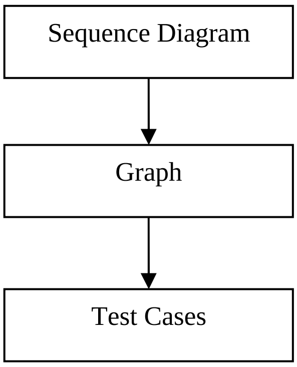

Fig. 1 Generation of Test Cases from UML Model

Fig 1 shows the overview of generation of test cases from UML model. Given a Sequence Diagram (SD), we transform into a graphical representation and Sequence Diagram Graphs (SDG). Each node in SDG stores necessary information for the test cases generation. This information are collected from the use case templates(also called extended use case), class diagrams and data dictionary expressed in the form of Object Constrained Language (OCL), which are associated with the use case for which the sequence diagram is considered. We then traverse SDG and generate test cases based on a coverage criteria and a fault model.

4.1. Transformation of an SD into SDG

The UML sequence diagram is transformed into a graph called the sequence diagram graph and augmenting the SDG nodes with different information necessary to compose test vectors. These information are mined from use case templates, class diagrams and data dictionary. The SDG is then traversed to generate the test cases. The test cases thus generated are suitable for system testing and to detect interaction and scenario faults.

4.2. Test Cases from Sequence Diagram

We use sequence diagram as a source of test case generation. A sequence diagram represents various interactions possible among different objects during an operation. A test set is therefore necessary to detect faults if any when an object invokes a method of another object and whether the right sequence of message passing is followed to accomplish an operation. From the SDG, it is evident that covering all paths from the start node to a final node would eventually cover all interaction as well as all messages sequence paths. We follow the coverage criterion stated below to derive the test set.

4.2.1. Coverage criteria

All sequence diagram message path sequence coverage criterion: given a test set T and a sequence diagram D, T must cause each sequence of message path exercise at least once. To generate test cases that specify the criteria, we first enumerate all possible paths from the start node to a final node in the SDG. Each path then would be visited to generate test cases. Every test strategy targets to detect certain category of faults called the fault model.

4.2.2. Generating test cases

The test criteria used and the process followed for sequence diagrams are described below.

4.2.2.1. Sequence diagram

In the UML, a message is request for a service from one UML actor to another. These are typically implemented as method calls. Each sequence diagrams represents a complete trace of messages during the execution of a user–level operation. We form message sequence path by using the messages and their sequence numbers. Message sequence paths can be the traces of system level interactions of component (object) level interactions.

4.2.2.2. Message Sequence Path Coverage

For each diagram in the specifications there must be at least one test case T such that when the software is executing using T, the software that implements the message sequence paths of the sequence diagrams is executed.

However, selection of test cases from number of test cases is the most challenging task. For this we propose a mining approach here.

5. Data Mining

In order to conduct effective data mining, one needs to first examine what kind of features and applied knowledge discovery system is expected to have and what kind of challenges on may face at the development of data mining techniques, handling of different types of data. Because there are many kinds of data and data bases used in different application, one may expect that a knowledge discovery system should be able to perform effective data mining on different kinds of data. Since most available databases are relational, it is crucial that a data mining system performs efficient and effective knowledge on relational data. Moreover many applicable data contains complex data types, such as structured data and complex data objects, hypertext and multimedia data, spatial and temporal data, transaction data and legacy data.

5.1. Applying Data Mining Concepts

There are many methods available for mining different kinds of data, including association rule, classification and clustering.

We can utilize any of these techniques based on [4]

- What kind of data bases to work on
- What kind of knowledge to be mined
- What kind of techniques to be utilized

5.1.1. Association

Association rules describe the association among items in the large data base. For example, one may find, from a large set of transaction data, such as association rule as if customer buys (one brand of) milk, he/ she usually buys (another brand of) bread in the same transaction. Using these association rules, we can derive the association patterns from large databases.

5.1.2. Data Classification

Data classification is the process which finds the common properties among a set of objects in a database and classifies them into different classes, according to a classification model.

5.1.3. Clustering

Clustering is the process of grouping the data into classes or clusters so that object within a cluster has high similarity in comparison to another, but is dissimilar to object in other clusters. Among all the mining techniques, clustering is the most effective technique which we are going to use for test case mining.

Clustering analysis helps constant meaningful partitioning of a large set of object based on a "divide and conquer" methodology which decomposes a large scale system into smaller components to simplify design and implementation. As a data mining task, data clustering identifies cluster or densely populated regions, according to some distance measurement, in a large, multidimensional data. Given a large set of multidimensional data points, the data space is usually not uniformly occupied by the data points. Data clustering identifies the sparse and the crowded places, and hence discovers the overall distributions patterns of the data set [1].

We can apply clustering technique for test case mining.

6. Intelligent Software Agent

Agent is authorized to act for another. Software Agent is an artificial agent which operates in a software environment. Intelligent Software Agent (ISA) is a software agent that uses Artificial Intelligence (AI) in the pursuit of the goals of its clients.

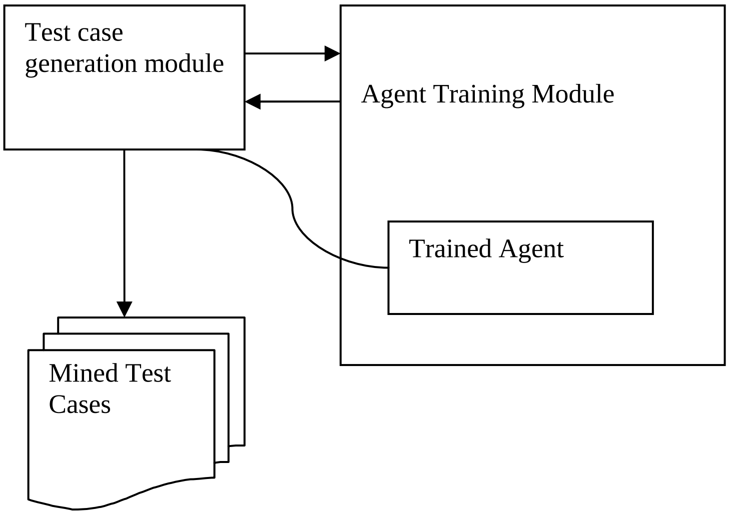

Fig 2 UML generated Test Case Mining using Intelligent Software Agents

To date, ISAs have been successfully employed in multiple application endeavors, some of which are listed below.

- Data Collection and Filtering
- Pattern Recognition
- Event Notification
- Data Presentation
- Planning and Optimization

- Rapid Response Implementation

Design of UML generated test case mining using ISA has 3 modules

- Test case generation module
- Agent training module
- Test case mining using ISA module

Fig 2 shows the mining of test cases using ISA. In Test case generation module the test cases are generated from sequence diagram. In Agent training module the agent is trained for mining the test cases. And then the trained agent is used.

7. Conclusion and Future Work

This paper has proposed a project on mining test cases using ISA which are generated from UML sequence diagram graph (SDG) and sequence diagram (SD) using mining techniques. The most important characteristics of test case mining are that the user can take out the required test cases from database without any human interactions. In future, we shall extend the data mining language to allow more flexible query specification, and develop and interactive data mining to test case mining.

8. References

[1] Zhijie Xu, Laisheng Wang, Jiancheng Luo, Jianqin Zhang. A Modified clustering algorithm for data mining. In: Geoscience and Remote Sensing Symposium, 2005. IGARSS apos;05. *Proc. 2005 IEEE InternationalVolume 2*, Issue , 25-29 July 2005 Page(s): 4 pp.

[2] Ming-Syan Chen, Jiawei Han, Philip S.Yu. Data Mining: An Overvies from a Database Perspective. *In: IEEE transactions on Knowledge and Data Engineering. Volume: 8*, Issue: 6on page(s): 866-883

[3] Dae-Kyoo Kim, Jon Whittle. Generating UML Models from Domain Patterns. In: Third ACIS Int'l Conference on Software Engineering Research, *Management and Applications (SERA'05)*, 2005. sera, pp.166-173,

[4] Tao Xie, Jian Pei, Ahmed E.Hassan. Mining software Engineering Data. *In: 29th International Conference on Software Engineering - Companion*, 20-26 May 2007. on page(s): 172-173

[5] Kansomkeat S, Offutt J, Abdurazik A, Baldini A. A Comparative Evaluation of Tests Generated from Different UML diagrams. In: Ninth ACIS International Conference on Software Engineering, *Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2008. SNPD '08. On page(s): 867-872

[6] J. Rajan, V. Saravanan. A Framework of an Automated Data Mining System Using Autonomous Intelligent Agents. *In: Proc of International Conference on Computer Science and Information Technology.* 2008. pp.700-704.

[7] Dymek D, Kotulski L. Using UML(VR) for Supporting the Automated Test Data Generation *In: Proc of Third International Conference on Dependability of Computer Systems.*June 2008. Page(s) : 3.