International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4 | ISSN : 2456-3307

Software Quality measurement with Metamorphic Testing

M. SatishKumar[1], S. Surekha[2], M. Keerthi[2]
[1]Assoc. Professor Department of computer Applications, SVCET, Chittoor, Andhra Pradesh, India
[2]PG scholar Department of computer Applications, SVCET, Chittoor, Andhra Pradesh, India

ABSTRACT

Web search engines are composed by thousands of query processing nodes, i.e., servers dedicated to process user queries. Metamorphic testing may be a testing technique which will be used to verify the useful correctness of software system within the absence of an ideal oracle. This paper extends metamorphic testing into a user-oriented approach to software system verification, validation, and quality assessment, and conducts large scale empirical studies with four major net search engines: Google, Bing, Chinese Bing, and Baidu. These search engines are very tough to check and assess using conventional approaches owing to the lack of an objective and generally recognized oracle. The results are useful for each search engine developers and users, and demonstrate that our approach will effectively alleviate the oracle drawback and challenges close a lack of specifications when verifying, validating, and evaluating giant and complex software systems.

Keywords: Metamorphic testing, Google, Bing, Chinese Bing.

I. INTRODUCTION

Computer-based application has been widely used all over the world. Hence, the roles of software systems have been increased exponentially. This causes, at the same time, the increasing reports of software faults. To guarantee the quality of software used is handled by software quality assurance process. It has become one of the most important areas in the software industry as well as in the academic sectors. Software testing, an important approach in software quality assurance, is widely reflected as a critical activity and now is one of main research focus in software engineering (Hailpern et al., 2002). One objective of software testing is to detect as quickly as possible, as many software faults as possible (Myers, 2004).

Software testing is one of phase in software engineering process that has a very improtant role to determine the quality of software under test. The general steps in software testing is generating test cases, selecting appropriate set of test cases based on certain criteria, executing them, and checking the outputs against a test oracle to determine whether any failures detected or not. A test oracle is a mechanism to check whether the output of executing a program under testing using a test case is according to the expected output or not. In other words, it is used to verify whether the progam has passed the test or not (Hierons, 2012). The presence of oracle testing is very important in conducting testing. However, in most situation, oracle testing is impractical to be found or too expensive which is known as an oracle problem (Manolache et al, 2001). Chen et al designed a new testing method, called Metamorphic Testing (MT) which was aimed to alleviate the oracle problem (Chen et al, 1998). This method is approached based on the property of program under test.

Based on the properties, tester is expected to generate some Metamorphic Relations that mainly have two functions: (i) to generate new test cases from the original test cases, and (ii) to verify whether test passes or fails based on the relations of the inputs and or outputs of original test cases and new test cases. This paper aims to introduce the use of MT in a case study of matrix multiplication. This case is chosen as matrix multiplication program can face oracle problem particularly when the size of matrices are large. However the case is quite common and widely used so that it will be easier to understand in explaining the concept used in MT.

II. METAMORPHIC TESTING

Metamorphic Testing (MT) is property bases testing which aims to find some useful relations (called Metamorphic Relations) to alleviate the oracle problems (Chen et al, 2003). As explained by Asrfai et al. (Asrafi et al, 2011), a metamorphic relation (MR) is an expected relation of the program under test which should be valid over a set of distinct input data and their corresponding output for multiple executions. Figure-1 sumarizes the relations in MT which involve source and follow-up inputs and outputs.MT checks the validity of MRs by multiply executing of the target program. The steps of MT are as folllowings: (i) determining specific properties of the SUT to construct MRs, (ii) generating source test case by some traditional testing techniques (such as random testing), (iii) generating follow-up test cases based on source test cases according to the MRs, (iv) executing the test cases, and (v) verifing the outputs of the test cases against MRs. If the outputs of the source and follow-up test cases do not match their relations in corresponding MR, then the test fails.

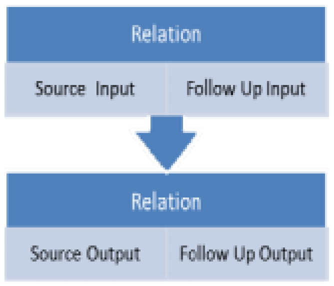


Figure-1. Relation in MT.

Asrafi et al (Asrafi et al, 2011) presented a simple example of MT in a sorting program as follows. The program sorts a set of integers in the ascending order. Suppose S is a set of numbers to be sorted. If the set S is rearranged in reverse order the output of the sorting program will still remain same. This MR can be denoted by Sort(S) = Sort (reverse(S)). Suppose S = {35, 15, 32, 25}, Sort(S) will yield {15, 25, 32, 35}.We reverse the set S to generate the follow-up test case reverse(S) = {25, 32, 15, 35}. If Sort (reverse(S)) {15, 25, 32, 35}, we can say a fault is detected. MT has been widely used in solving many oracle problems (Barus et al, 2009; Chen et al, 1998; Chen et al, 2009; Chen et al, 2004).

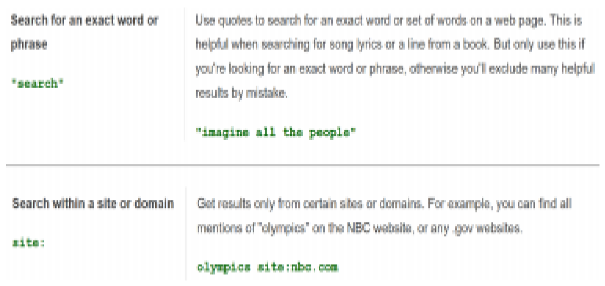
Proposed system:-

To apply MT to the automatic quality assessment of search engines, without the need for an oracle or human assessor, two groups of MRs were used: The "No Missing Web Page" group assesses the search engines' capability in retrieving appropriate Web pages to meet the users' needs; and the "Consistent Ranking" group assesses the ranking quality of the search results. This section provides a brief description of these MRs.

Metamorphic Relation: MPSite

MPSite belongs to the "No Missing Web Page" group of MRs, which assess the search engine's Web page retrieval capability. MPSite is focused on the search engine's reliability when retrieving Web pages that contain an exact word or phrase. It therefore assesses the keyword based search feature. MPSite is described as follows: Let A be a source query for which the search engine returns a non-empty list of results (called the source response), namely, (p1, p2, . . . , pn), where $0 < n$ and pi is a Web page from domain di , $1 \leq i \leq n$. To enhance accuracy and validity of our approach, in MPSite we only consider situations where $0 < n \leq 20$ so that we can avoid the inaccuracy associated with large result sets (such as a large list being truncated by the search engine to improve response time).

For the source response (p1, p2, . . . , pn), n follow-up queries are constructed as follows: The ith follow-up query Bi (1 ≤ i ≤ n) is constructed in such a way that Bi is identical to A except that Bi includes an additional criterion which requires that all results be retrieved from domain di . Let FRi (a follow-up response) be the list of Web pages returned by the search engine for query Bi . The metamorphic relation MPSite requires that pi ∈ FRi (note that there is no requirement on the ranking of pi in FRi). For example, let us test Google by issuing the following source query:



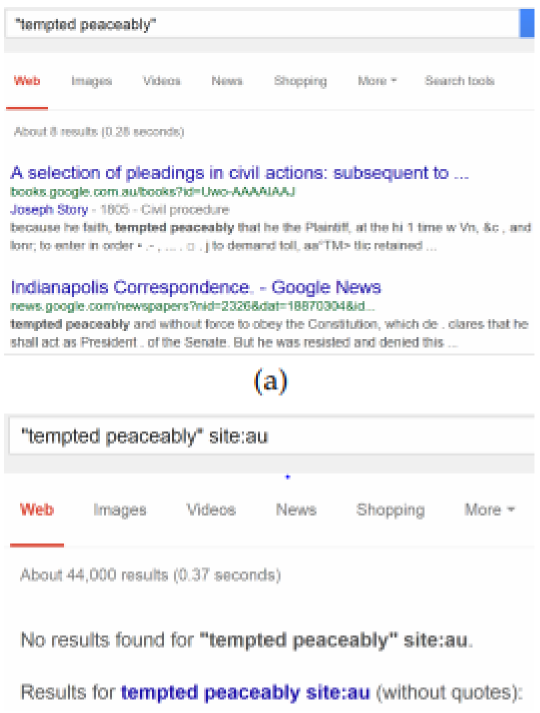
Search for an exact word or phrase "search"	Use quotes to search for an exact word or set of words on a web page. This is helpful when searching for song lyrics or a line from a book. But only use this if you're looking for an exact word or phrase, otherwise you'll exclude many helpful results by mistake. "imagine all the people"
Search within a site or domain site:	Get results only from certain sites or domains. For example, you can find all mentions of "olympics" on the NBC website, or any .gov websites. olympics site:nbc.com

Figure 2. Excerpts from Google help page.

"side effect of antibiotics in babies" where the quotation marks are part of the query. Google returned a total of 7 Web pages. Without loss of generality, let us consider the top result, which is: This Web page is from the .uk domain. 1 The metamorphic relation MPSite enables the construction of the following follow-up query: ["side effect of antibiotics in babies" site:uk], 2 where “site:” is a Google search operator that specifies domains (see Figure 2 (lower)). Obviously, the previously returned top result (http://www.dailymail.co.uk/. . .) meets this search criterion, is indexed in Google database, and therefore should still be returned by Google for this follow-up query. In this example, Google returned a total of 7 Web pages for the source query. Therefore, 7 followup queries are constructed by referring to MPSite. 3 Using MPSite, even if the assessor is unable to verify or evaluate each individual response, he/she can still verify the logical consistency relationship among multiple responses. Here, the basic approach is to use the search engine’s source response to check its follow-up response. Figure 3 shows a failure detected using MPSite,All MRs identified in this paper were implemented into a testing tool and, hence, the testing and assessment process is automated.

Metamorphic Relation: MPTitle:-

For many search engines including those investigated in the present paper, if the words are not enclosed by double quotation marks, synonyms will be employed automatically. For instance, Google specifies that “Google employs synonyms automatically, so that it finds pages that mention, for example, childcare for the query [child care] (with a space), or California history for the query [ca history].” Synonyms are employed because the search engines attempt to return Web pages that best meet users’ information needs. In other words, the search engines attempt to imitate the behavior of a human operator, to which end, correct understanding of the Web pages and of the user intent are key. To test a search engine’s information retrieval capability in situations where synonyms may be used for semantic search, a good strategy is to construct a test query q that best characterizes a target Web page p (the words in q may or may not directly appear in p). Furthermore, p must have been indexed in the search engine’s database. The search engine can be tested on q. If p is not retrieved, then the user’s perception of the search quality will be poor. A research question is: How can such q’s and p’s be identified in a fully automatic fashion for search engine assessment? The metamorphic relation MPTitle is designed to meet this challenge.


"tempted peaceably"

About 8 results (0.28 seconds)

A selection of pleadings in civil actions: subsequent to ...
books.google.com.au/books?id=Uwo-AAAAIAAJ
Joseph Story - 1805 - Civil procedure
because he faith, **tempted peaceably** that he the Plaintiff, at the hi 1 time w Vn, &c , and lonr; to enter in order • .- , ... □ . j to demand toll, aa"TM> tlic retained ...

Indianapolis Correspondence. - Google News
news.google.com/newspapers?nid=2326&dat=18870304&id...
tempted peaceably and without force to obey the Constitution, which de . clares that he shall act as President . of the Senate. But he was resisted and denied this ...

(a)

"tempted peaceably" site:au

About 44,000 results (0.37 seconds)

No results found for **"tempted peaceably" site:au**.

Results for **tempted peaceably site:au** (without quotes):

Figure 3. A Google failure detected using MPSite. The top result in (a) cannot be retrieved in (b).

Metamorphic Relation: MPReverseJD:-

The third MR of the "No Missing Web Page" group is MPReverseJD. Its design was inspired by a search engine assessment technique informally used in industry, which is based on the rationale that a good search engine should return similar results for similar queries. For instance, although a search for [today's movies in Redmond] and a search for [Redmond movies today] (without double quotes) may return different results, the two result sets should share a large intersection if the search engine is robust to the nonessential differences between these two queries. 5 This idea was also employed by Imielinski and Signorini to test semantic search engines using semantically equivalent queries [29]. The MR MPReverseJD is designed as follows: The source query A is defined to be a query for which the search engine returns a non-empty list of up to 20 results. A is further defined to be the conjunction of up to 4 terms, namely:

$$A = A_1 \text{ AND } A_2 \underbrace{\text{AND } A_3}_{optional} \underbrace{\text{AND } A_4}_{optional}$$

where Ai (i=1, 2, 3, 4) is a name enclosed by double quotation marks. Terms A3 and A4 are optional: A3 is applied only when the conjunction of A1 and A2 has more than 20 results, and A4 is applied only when the conjunction of A1, A2, and A3 has more than 20 results. If the conjunction of all 4 terms still has more than 20 results, all these terms will be discarded and a new query will be formed. The following is an example of the source query A: ["Vincent Van Gogh" AND "Elvis Presley" AND "Albert Einstein" AND "Plato"]. In this example, A1="Vincent Van Gogh," A2="Elvis Presley," A3="Albert Einstein," and A4="Plato." The follow-up query B is constructed by reversing the order of A's terms: ["Plato" AND "Albert Einstein" AND "Elvis Presley" AND "Vincent Van Gogh."] MPReverseJD states that a stable search engine should return similar results for the source query A and followup query B.

In other words, the two result sets should have a large intersection – we refer to this kind of quality as stability. This requirement is reasonable especially given that the result set of A is very small (containing no more than 20 results) and that the source and follow-up queries have similar semantic meanings – this is because the queries only consist of names whose orders do not change the meaning of the queries. To measure the similarity of the two result sets, we use the metric Jaccard similarity coefficient (or Jaccard coefficient for short), defined as |X|/|Y |, where X = source response ∩ follow-up response and Y = source response ∪ follow-up response. The source and follow-up responses refer to the source and follow-up queries' result sets, respectively. Obviously, 0 ≤ Jaccard coefficient ≤ 1. A larger Jaccard coefficient indicates higher similarity and, hence, better stability. Given that the vast majority of users would prefer stable search results, poor stability may result in a poor user experience. (In this paper, "user experience" refers to users' perceived quality of the search results.)

Metamorphic Relation: SwapJD:-

The second group of MRs is named "Consistent Ranking." Its first MR is SwapJD, which assesses the search engines' ranking stability based on the concept that a stable search engine should return similar results for similar queries. SwapJD is described as follows: The source query A contains only two words (without quotation marks) and the follow-up query B is constructed by swapping the two words. A stable search engine should return similar results for A and B if these two queries have similar meanings, regardless of their word orders. The similarity can be measured by calculating the Jaccard coefficient of the top x results in the two result lists, where x can be given by the assessor. In this research, we set x to 50, as our experience suggests that most users are unlikely to browse search results beyond the top 50.

Metamorphic Relation: Top1Absent:-

The Top1Absent MR focuses on the ranking quality of the very first result presented in the search results screen. This top result can be considered as the most important one among all search results. According to Imielinski and Signorini [29], more than 65% of search clicks are done on the first result. Top1Absent is designed by extending the idea of MPSite, as described below: The source query A is a word randomly selected from an English dictionary (excluding common words such as "is" and "of") and is surrounded by double quotes. Let p1 be the top result, that is, p1 is the first listed Web page returned by the search engine for query A. The follow-up query B still uses A as the query term, but restricts the search to p1's domain only. The expected relationship is that p1 should still appear in the search results for B.

III. CONCLUSION

Metamorphic testing (MT) was at the start proposed as a verification technique, wherever metamorphic relations (MRs) were identified by referring to the target algorithmic rule to be enforced. During this paper, we have demonstrated the practicability of MT being a unified framework for software verification, validation, and quality assessment. We have a tendency to conduct a study on search engines, where we have a tendency to known MRs from the users' perspective without bearing on the target algorithms or system specifications. more generally, this approach permits users to recognize whether or not or not a system is suitable for their specific wants within the absence of complete software documentation, that is usually the case with net services, poorly evolved software, and open source software.

IV. REFERENCES

[1]. G. Agha. Actors: a model of concurrent computation in distributed systems. MIT Press, 1986.

[2]. J. Aldrich, C. Chambers, and D. Notkin. Archjava: connecting software architecture to implementation. In Proceedings of the 24th international conference on Software engineering, pages 187-197. ACM Press, 2002.

[3]. J. Aldrich, V. Sazawal, C. Chambers, and D. Notkin. Language support for connector abstractions. In ECOOP 2003 - Object-Oriented Programming: 17th European Conference, volume 2743 of Lecture Notes in Computer Science, pages 74-102. Springer-Verlag, July 2003.

[4]. G. Arango, L. Bruneau, J. F. Cloarec, and A. Feroldi. A tool shell for tracking design decisions. IEEE Software, 8(2):75-83, March 1991.

[5]. The Archium website, http://www.archium.net.

[6]. M. Babar, I. Gorton, and B. Kitchenham. A framework for supporting architecture knowledge and rationale management. In A. H. Dutoit, R. McCall, I. Mistrik, and B. Paech, editors, Rationale Management in Software

Engineering, chapter 11, pages 237-254. Springer-Verlag, March 2006.

[7]. M. A. Babar, R. C. de Boer, T. Dingsøyr, and R. Farenhorstir. Architectural knowledge management strategies: approaches in research and industry. In Proceedings of the 2nd Workshop on SHAring and Reusing architectural Knowledge - Architecture, rationale, and Design Intent (SHARK/ADI 2007), May 2007.

[8]. M. Bachler, S. Buckingham Shum, D. D. Roure, D. Michaelides, and K. Page. Ontological mediation of meeting structure: Argumentation, annotation, and navigation. In 1st International Workshop on Hypermedia and the Semantic Web, 2003.

[9]. E. L. A. Baniassad, G. C. Murphy, and C. Schwanninger. Design pattern rationale graphs: Linking design to source. In Proceedings of the 25th ICSE, pages 352-362, May 2003.

[10]. L. Bass, P. Clements, and R. Kazman. Software architecture in practice. Addison Wesley, 1998.

[11]. L. Bass, P. Clements, and R. Kazman. Software architecture in practice 2nd ed. Addison Wesley, 2003.

[12]. L. Bass, P. Clements, R. L. Nord, and J. Stafford. Capturing and using rationale for software architecture. In A. H. Dutoit, R. McCall, I. Mistrik, and B. Paech, editors, Rationale Management in Software Engineering, chapter 12, pages 255-272. Springer-Verlag, March 2006.

[13]. D. Batory, J. Liu, and J. N. Sarvela. Refinements and multi-dimensional separation of concerns. In Proceedings of the 9th European software engineering conference, pages 48-57. ACM Press, 2003.

[14]. K. H. Bennett and V. T. Rajlich. Software maintenance and evolution: a roadmap. In Proceedings of the conference on the future of Software engineering, pages 73-87. ACM Press, 2000.

[15]. B. W. Boehm, E. Horowitz, R. Madachy, D. Reifer, B. K. Clark, B. Steece, A. W. Brown, S. Chulani, and C. Abts. Software Cost Estimation with Cocomo II. Prentice Hall, January 2000.

[16]. C. Boekhoudt. The big bang theory of ides. Queue, 1(7):74-82, 2003.

[17]. G. Booch, J. RumBaugh, and I. Jacobson. The unified modeling language user guide. Addison Wesley, 1998.

[18]. J. Bosch. Superimposition: A component adaptation technique. Information and Software Technology, 41(5):257-273, 25 March 1999.

[19]. J. Bosch. Design & Use of Software Architectures, Adopting and evolving a product line approach. ACM Press/Addison Wesley, 2000.

[20]. J. Bosch. Maturity and evolution in software product lines: approaches, artefacts and organization. In Proceedings of the 2nd Software Product Line Conference (SPLC 2002), August 2002.

Author's Profile:

M. SatishKumar M.C.A., M.Tech., M.phil., working as an Assoc.professor in Sri Venkateswara college of engineering &technology, Chittoor, A.P.

S. Surekha received the PG degree from SriVenkateswara college of engineering & Technology, Chittoor, A.P.

M. Keerthi received the PG degree from Sri Venkateswara college of engineering& technology ,Chittoor, A.P.