

Component Based Software and test case prioritization in component based software

Neetu Khanna, Suman deswal

Abstract— Software systems evolve into complex systems every day. Component-based software model is a new trend in software development. The main idea is to reuse already completed components instead of developing everything from the very beginning each time. Use of component-based software development brings many advantages: faster development, lower costs of the development, better usability, etc. Software engineers usually use many test suites to detect whether faults introduced into Component Based models. The testing technique works by to doing all existing test cases, but technique however require a lot of time and efforts, depending on the size and complexity of the component based system under test. This paper reviews test case prioritization in CBS models.

Index Terms—Test Cases, Component Based Software Engineering, Test case prioritization.

I. INTRODUCTION

Component Based Software (CBS) System are mainly constructed from reusable components such as third party components and Commercial-Of-The-Shelf (COTS) components with minimum engineering efforts and resource cost and to facilitate fast-paced delivery of scalable, evolvable software systems. Software applications are assembled from components from a variety of sources; the components may be written in several different programming languages and run on several different platforms or we can say that components are heterogeneous in nature [1]-[2]. A Software component is a software building block that is conforms to a component model & it can be independently composed and deployed without modification according to the composition standard [2]. By using components it is possible to produce more functionality with the same investment of time and money [3]. A generally accepted view of a software component is that, it is a software unit with provided services and required services from others too (Fig. 1). The provided services are operations performed by the component whereas, the required services are the services needed by the component to provide target services [4].

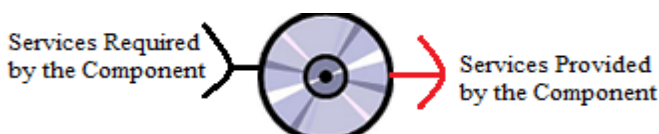


Fig.1 Service Scheme of Software Component [4].

II. COMPONENT BASED SOFTWARE ENGINEERING

Component based software engineering is widely accepted as a new and latest approach to software development. Nowadays the software systems are highly complex, large and uncontrollable. These result in lesser productivity, higher risk management and greater software quality. So, there arises the need for component based systems [5]. Component-based software engineering (CBSE) intends to build large software systems by integrating pre-built software components. The high productivity is achieved by using standard components [6].

The principles of CBSE can be best described by the following two guiding principles: reuse but do not reinvent; assemble pre-built components rather than coding line by line [6].

A component:

- Independent and replaceable part of the system;
- Performs a particular function;
- Works on a well-defined architecture;
- Communicates with the help of interfaces [5].

CBSE is a process of integrating the various software components to form an application to satisfy a functionality [5]. CBSE shifts the emphasis from programming to composing software systems [4].

A. Goal of CBSE:

- Cost reduction;
- Ease of assembly;
- Reusability;
- Maintainability;
- Customization and reusability [7].

III. COMPONENT BASED SOFTWARE DEVELOPMENT

Every product, including software products, has a lifecycle. A products lifecycle phases are: concept, development, production, utilization and retirement [8].

Modern software systems become more and more large-scale, complex and uneasily controlled, resulting in high development cost, low productivity, unmanageable software quality and high risk to move to new technology. So growing demand of searching for a new, efficient and cost-effective software development paradigm [9]. There exist many models for software development processes and lifecycles. Most of them are specified considering some specific, often non-technical goals, such as quality, predictability, dependability, or flexibility, and are often independent of technology [8]. CBSD is based on the idea that software

systems can be developed by selecting appropriate off-the-shelf components (COTS) and then assembling them with a well-defined software architecture. Where COTS components can be checked out from a component repository, and assembled into a target software system as shown in (Fig. 2) [9].

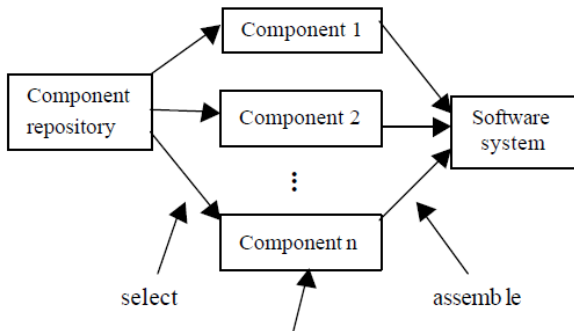


Fig.2 Commercial Off-the-shelf (COTS) components [9].

A. Component based software architecture

CBSE emphasizes modular architecture so that we can partially develop a system and incrementally enhance the functions by adding and/or replacing components. To make such design possible, we need a sound foundation of software systems, that is, software architecture. They are provided in the form of frameworks. Frameworks are workable reference to the underlying software architecture. To be effective, framework can be hierarchical up from domain independent to domain specific [10].

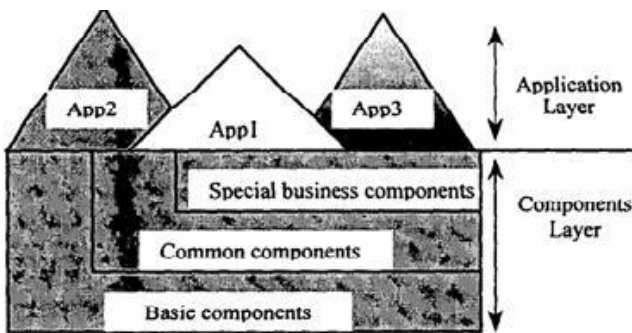


Fig.3 Component Based System Architecture [2].

Software architecture is a high level abstraction of a software system: its components and their connections. Thus, architecture complements component definition which focuses on the individual components and their interfaces [11].

Advantages of component based software systems over the traditional systems are:

- Less time consuming.
- Components can be integrated with ease.
- Lesser cost incurred in building.
- Lesser time-to-market.
- Follows predictable architectural patterns.
- Provides higher quality results.
- Increases reusability.

- Increases maintainability [5].

IV. TEST CASE PRIORITIZATION TECHNIQUES

Test case prioritization techniques schedule test cases in an execution order according to some criterion. Because running all test cases in an existing test suite can consume an inordinate amount of time. So testers may want to order their test cases so that the test cases with the highest priorities, will run first to maximize some objective function. When the time required to execute all test cases in a test suite is short, test case prioritization may not be cost effective [12].

- A. Ant Colony Optimization based on foraging behavior of ants in nature. During food hunting, ants leave pheromone on the traveled paths, and the shortest path will be discovered through teamwork and pheromone evaporation process. In beginning ants choose random paths toward the food source (Fig.4). Pheromone evaporates in different evaporation rates according to the length of paths, therefore, the longer the path, the more pheromone evaporation. The other follower ants will choose the shorter path (Fig.5) according to the amount of residual pheromones. Through the time evolution of the group cooperation, ants will eventually choose the shortest path. In addition, the number of ants also affects the search result of the shortest path. In the initial stage of foraging search, there must be a sufficient number of ants to perform path mining. If the ant colony size is too small, it would not be able to comprehensively search for all the possible paths and may lead into a local optimization [13].

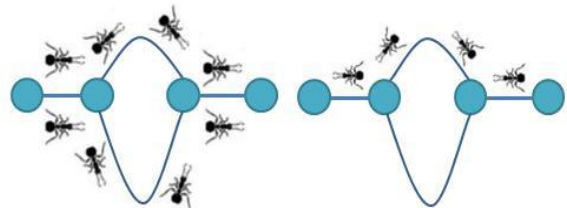


Fig.4 Random Paths. Fig.5 A shortest path.

- B. In test case prioritization “Quality based requirement” is an innovative approach for evaluating the quality of requirements in software projects based on multiple quality evaluation criteria. Specifically, it presents a methodology that uses Desirability Functions to create a unified measurement that represents how well requirements meet quality attributes and how important the quality attributes are for the project. By modifying the parameters of the desirability functions, quality and priority of requirements can be evaluated. Overall, this approach proved to be a feasible technique for

efficiently evaluating the quality and priority of requirements in software projects [14].

- C. Genetic Algorithm is based on the idea on the natural evolution and lies on the concept of the survival of fittest into a solution space. Each cycle of GA process includes initialization (encoding), selection based on fitness function, reproduction using crossover or mutation. The cycle is repeated till a solution is found [15]. Genetic algorithm collect pairwise information useful to prioritize the test case for a software system. They have applied algorithm to a real case study, consisting of a non trivial number of requirements, which makes AHP hardly applicable. This approach scaled to the size of the considered case study and produced a result that outperforms GA (i.e., a genetic algorithm which optimizes satisfaction of the initial constraints, without gathering further constraints from the user). Specifically, by eliciting between 50 and 100 pairwise comparisons from the user it was possible to obtain a substantially better ordering of the prioritized requirements [16].

V. CONCLUSION

In this paper we study about component based software engineering and some test case prioritization technique.

Component based software development qualifies, adapts, and integrates software components for reuse in a new system. A component is a trivial component is a non-trivial, nearly independent and replaceable part of a system that fulfills a clear function in the context of a well-defined architecture. When a software system is developed then to ensure the quality, reliability, robustness and functionality of system testing is necessary. Test case prioritization is a method of scheduling and ranking the test cases from multiple test suites of software to reduce time required for testing.

ACKNOWLEDGMENT

I would like to give my heart-felt sincere-most thanks to all lab faculty members who always stood as a guiding spirit behind me and gave their invaluable supervision, guidance and support without which this project wouldn't have reached its final destination.

REFERENCES

- [1] Neelam Sirohi, Anshu Parashar, "Component Based System and Testing Techniques", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 6, June 2013,ISSN(print) : 2319-5940,ISSN (Online) : 2278-1021.
- [2] Ye Wu, Dai Pan and Mei-Hwa Chen, "Techniques for Testing Component-Based Software", 2001 IEEE, 1050-4729/01\$10.00. J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34-39, Jan. 1959.
- [3] Heineman, G.T. and W.T.councill, "Component-Based Software Engineering:putting the piece together",Reading MA: Addison-weseley, 2001.
- [4] Heineman, G.T. and W.T.councill, "Component-Based Software Engineering: putting the piece together",Reading MA: Addison-weseley, 2001.
- [5] Ardhendu Mandal, S.C.Pal, "Emergence of Component Based Software Engineering", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 3, March 2012, ISSN: 2277 128X.
- [6] Divya Chaudhary, "Component Based Software Engineering Systems: Process and Metrics", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013, ISSN: 2277 128X.
- [7] Ju AnW ang, "Towards Component Based Software Engineering", University of Nebraska at Kearney, Kearney, NE 68849, USA.
- [8] Wilhelm Hasselbring, "Component Based Software Engineering", International Journal of Software Engineering and knowledge Engineering, vol. 0, No. 0 (0000) 000-000.
- [9] Wilhelm Hasselbring, "Component Based Software Engineering", International Journal of Software Engineering and knowledge Engineering, vol. 0, No. 0 (0000) 000-000.
- [10] Xia Cai, Michael R. Lyu, Kam-Fai Wong, Roy Ko, "Component-Based Software Engineering: Technologies, Development Frameworks, and Quality Assurance Schemes", The Chinese University of Hong Kong, Hong Kong Productivity Council.
- [11] Mikio Aoyama, "New Age of Software Development: How Component-Based Software Engineering Changes the Wayof Software Development ?", Department of Information and Electronics Engineering Niigata Institute of Technology.
- [12] Gregg Rothermel,Roland H. Untch,Chengyun Chu,Mary Jean Harrold, "Test Case Prioritization: An Empirical Study" Proceedings of the International Conference on Software Maintenance, Oxford, UK, September, 1999, IEEE Copyright.
- [13] Chien-Li Shen, Eldon Y. Li, "Apply ant colony algorithm to test case prioritization",Department of Information Management, College of Commerce National Chengchi University, Taiwan
- [14] Carlos E. Otero, Erica Dell, Abrar Qureshi, Luis D. Otero, "A Quality-Based Requirement Prioritization Framework Using Binary Inputs", IEEE 978-0-7695-4062-7/10 \$26.00,DOI 10.1109/AMS.2010.48.
- [15] Bharti Suri, Isha Mangal, "Analyzing Test Case Selection using Proposed Hybrid Technique based on BCO and Genetic Algorithm and a Comparison with ACO", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012, ISSN: 2277 128X.
- [16] Paolo Tonella, Angelo Susi, Fondazione Bruno Kessler,Francis Palma, "Using Interactive GA for Requirements Prioritization", IEEE 2nd International Symposium on Search Based Software Engineering, 978-0-7695-4195-2/10 \$26.00 , DOI 10.1109/SSBSE.2010.17.

Authors Details:

1) Name: Neetu khanna

Education: M. Tech (2013-2015), Dept. of Computer Science and Engg., DCR University of Science and Technology, Sonapat(Haryana), India.

2)Name:Suman Deswal

Education: Dept. of Computer Science and Engg., DCR University of Science and Technology, Sonapat, India. Suman Deswal has completed her M.Tech (Computer Science and engineering) from Kurukshetra University, Kurukshetra, India and B.Tech (Computer Science & Engineering) from DCRUST, Murthal, India (erstwhile CRSCE Murthal) in 2009 and 1998 respectively. Currently, She is enrolled for Ph.D. and working on mobility in Heterogeneous Networks.

Job detail: Assistant professor, Dept. of Computer Science and Engg., DCR University of Science and Technology, Sonapat(Haryana), India.

She has contributed in various research papers in International journals:

- [1]Suman Deswal, Sukhbir Singh, "Implementation of Routing Security Aspects in AODV", International Journal of Computer Theory and Engineering, Vol. 2, No. 1 February, 2010, 1793-8201.
- [2]Suman Deswal, Anita Singhrova, "A comparative analysis of number portability routing schemes", International Journal of Computer Networks & Communications (IJCNC) Vol.6, No.2, March 2014.