Optimizing Code Reusability for Mobile Applications

V.Priyadharshini
Research Scholar,
School of Computer Science and Engineering,
Bharathiar University, Coimbatore,
Tamil Nadu, India.

ABSTRACT

Recent optimization technique in mobile application is used for games application, to reduce the memory space. In this research, a set of test cases and optimization techniques are used to reduce the memory space in mobile application. The focused features are message, contacts and calculator. A set of test scenarios are created for determining the quality and completeness of the proposed software which is experienced with GUI (Graphical User Interface).

The emerged applications have various performance characteristics and bottlenecks, because performance varies across different hardware, operating systems, and mobile. The research perspective is to promote awareness of Java performance issues. So that the user can make appropriate design and implementation choices for specific applications. The performance of mobile software analysis generates a report for the memory which is used by an application. If the memory is not sufficient to load the software, the proposed software finds the lengthy names by generating test scenario. By optimizing java constructors, packages, methods, classes, unused variables and loops in the application, the execution time of the application is reduced. Finally the statistical report of the optimized value is evaluated.

Keywords: Test Cases, Mobile Application Techniques, Validation Testing, Reusability, Optimization.

1. INTRODUCTION

Mobile applications allow the user to control or filter the information flow and communication through the devices. Mobile connectivity improves collaboration via real-time or instant interactivity, regardless of time and location, leading to better decision making. Mobile connectivity enhances customer orientation as users have better access to the service providers and do a better job in balancing the work life through a productive use of time [2].

Applications involving communications over a network use traditional client-server paradigm in which a connection is established between the client and the server or datagram's are sent across the network. To overcome the user's inability of mobile usage, many researches have identified with Mobile applications have become very high volume products, where cost and low power have been the main design criterion. The features of a phone are available through applications that use services offered by servers. Applications can use the services of all servers, and servers can use services belonging to other servers in order to provide the requested functionality [3].

2. LITERATURE REVIEW

The existing variety of mobile technologies, platforms and devices cause additional difficulties in developing and testing mobile software. The proposed GUI tests check the user interface and functionality of the mobile application.

To overcome user's inability of mobile usage, many researches have identified with mobile applications that it has very high volume products, where cost and low power are the main design criteria.

Mobile applications generally allow the user to control or filter the information flow and communication through the device is usually personalized or individualized.

Mobile services differ from traditional systems in that the mobile services are ubiquitous, portable and can be used to receive and disseminate personalized and localized information.

It is dynamic that it makes use of instrumentation of the SUT (System under Test) and run-time verification technology. The starting point of the method is a test-input generator, which generates inputs to SUT. These inputs are also fed to a report generator, which generates reports that the SUT must satisfy on these particular inputs.

The test-input generator and the report generator are specifically written for the application to be tested. It has a lot of similarities to test-case generation, since the system under test must be run a number of times, with different inputs, to obtain a set of traces.

3. PROPOSED WORK

Mobile devices are becoming smaller and smaller, but with faster processing times and larger storage capacity. Corresponding mobile interfaces also need to be modified in order to suit the requirements of new business models and enhance the mobile usability.

GUI mobile application must be designed such that the user must efficiently use mobile contents. The testing process must be verified to make sure that the process is working in an error-free manner.

The hardware usability issues in mobile phones are considered, in order to form design guidelines and eventually design proposal focusing on high usability. The goal with validation testing is typical to ensure that the product meets the usability requirements. The mobile devices, monitoring scheme collects and analyzes dynamic information of mobile applications, such as CPU, memory usage, etc.

The requirements improve quality and efficiency in the testing process of an application. To optimize the development usage in mobile applications, it is necessary to consider the mobile usage context for the design and the evaluation of the user system interaction of a mobile application.

Proposed mobile context factors and requirements are common basis for both inspection and user test. After conducting analysis and user test, the results described are mapped and discussed. The mobile context identified defines and describes the usage context of a mobile application.

4. TEST CASES

A test case is a set of conditions or variables and possible inputs that are developed for a particular goal or objective to be achieved on a certain application to judge its capabilities or features.

The mechanism for determining whether a software program or system has passed or failed is known as a test oracle. Test cases are often referred to as test scripts, test scenarios or test suites. One of the core problems of testing is the infinity of possible tests. Test scenarios are written by testers to address the testing needs of the mobile application. Test cases are derived from test scenarios which are related to requirements and designs [4].

In this research, the selection of test cases determines the type and scope of the test. The test case design determines the quality of a test. Therefore the research mainly focuses on test activities that exist before and after test execution. Activities such as planning and control, choosing test conditions, designing test cases and checking results, evaluating completion criteria, reporting on the testing process and system under test, and finalizing or closure [8].

Testing is widely used in industry for quality assurance. Testing is one of the software management practices that helps improve predictability and consistency. The most visible part of testing is executing tests. But to be effective and efficient, test plans should also include time to be spent on planning the tests, designing test cases, preparing for execution and evaluating status. Test execution is just one way to discover defects earlier in the development process [5].

The effectiveness of finding defects by testing and reviews can be improved by using independent testers. The test plan should include both positive and negative scenarios to demonstrate that the software reacts acceptably with good or bad data.

Fig.1 describes test case generation process.

In Development- In this phase, tester writes test cases for particular application.

Awaiting Review- In this phase, tester has completed test case, it must be reviewed. If the test case is accepted, the written test case is executed. If test case is rejected, required changes are made and reviewed again.

Awaiting Run- Here, test case is run, if it executes successfully, test case status is passed. If the execution failed, defect must be corrected to fix issue.

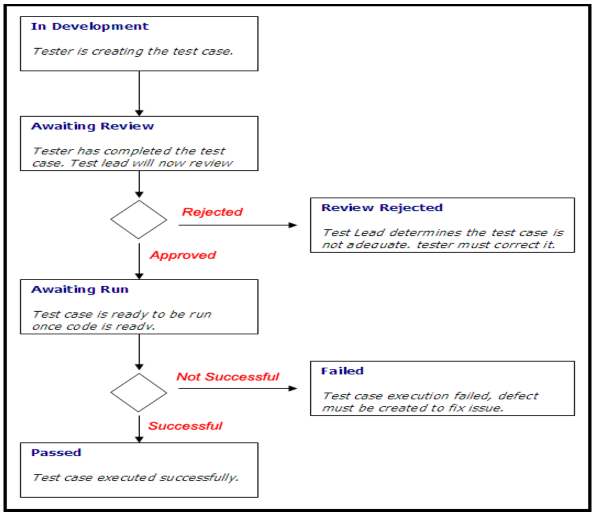
In Development
Tester is creating the test case.
Awaiting Review
Tester has completed the test case. Test lead will now review
Rejected
Review Rejected
Test Lead determines the test case is not adequate. tester must correct it.
Approved
Awaiting Run
Test case is ready to be run once code is ready.
Not Successful
Failed
Test case execution failed, defect must be created to fix issue.
Successful
Passed
Test case executed successfully.

Fig.1 Test Case Generation

Issues in Test Case

i) Readability of test cases: The semantic gap between the test scripts and the test tasks automated by the scripts is small. It is easy to read a test script and understand what GUI feature the script is designed to test.

ii) Platform independence: Regardless the platform a GUI application is developed on, GUI Test can be used to test the GUI's visual feedback.

iii) Separation of design and implementation: Test cases can be generated by designers and handed to programmers to implement features that must pass the test cases, to eliminate the biases that may arise when programmers are asked to test their own implementation.

5. MOBILE APPLICATION TECHNIQUES

Wireless data communications in form of Short Message Service (SMS) have gained global popularity, yet, not much has been done to extend the usage of these devices in information sharing. This framework provides the requirements to develop application that can be used to share information between client and server simulated mobile. A prototype application has been developed to demonstrate the important functionality of the proposed system in simulated environment [7].

The server-client architecture allows applications to run on the server window. The J2ME application can be hosted on as many mobile devices as required. Each of the mobile phone will act as SMS delivery component. Systems have recently emerged as a key technology in mobile computing systems for increasing both data rates and system performance [1].

a) Portability

In the face of hardware and software diversity of today's handheld devices, portability inevitably becomes a puzzle for mobile application developers. Portability primarily depends on (a) runtime support, and (b) the feasibility of achieving identical cross-platform look-and-feel and functionality. It is fragmented with respect to developing cross-platform applications.

b) Functionality

The aim of implementing multimedia-rich full-fledged applications is better served through Java ME due to the numerous APIs implemented.

c) Development speed

The safest way to guarantee steep learning curve and higher development speed is to take advantage of the developers' programming experience on the implementation of desktop applications.

d) Performance

Mobile applications are becoming increasingly computationally heavy, thereby raising requirements for increased runtime speed and storage. Performance metrics like processing overhead, memory consumption, and frame rate and deployment file size also depend on the particular development platform toolset.

e) Optimization

It is a relatively recent optimization technique of the swarm intelligence paradigm. Simple methods were developed for efficiently optimizing non-linear mathematical functions.

The hardware usability issues in mobile phones are considered, in order to form design guidelines and eventually design proposal focusing on high usability. The goal with validation testing is typical to ensure that the product meets the usability requirements. The mobile devices, monitoring scheme collects and analyzes dynamic information of

6. VALIDATION TESTING

Validation testing is done to ensure that all the inputs given are validated and generated. This test is typically responsible for the programmer to transform the design into implementation. It runs only with the simulator, not with real devices, it lacks an automated test Applications written for mobile devices have become more and more complex and sophisticated, adjusting to the constantly improving computational power of hardware. With the growing application size the need for automated testing frameworks, particularly frameworks for automated testing of user interaction and graphical user interface. While such testing has been thoroughly discussed in literature with respect to desktop applications, mobile development limits the possibilities significantly. To the best knowledge only a few solutions for creating automated tests of mobile applications exist and their functionality is very limited in general or constrained to only proprietary devices.

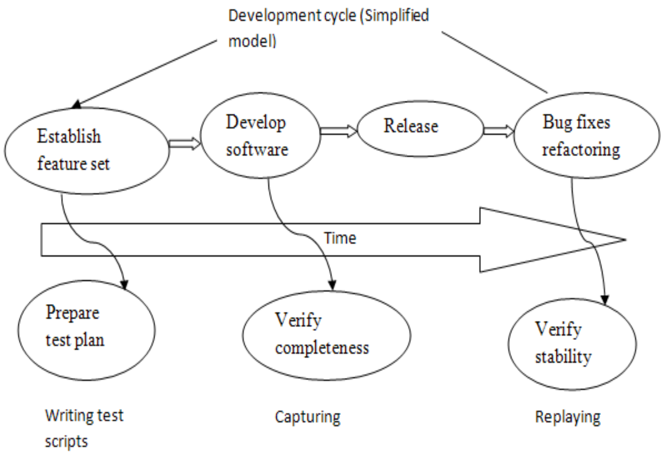

Fig.2 Development stages and Testing steps

Fig.2 describes the development cycle of the project requirements and main features for the proposed application. It describes a variety of tasks or activities that take place during the development process. The developer investigates the need for possible software automation for the given system. A feature set is a group of function, where the enhancements and new capabilities are determined and planned for development of a mobile application.

Developing software describes the development of a software product in a planned and structured process. It is known as software development cycle. Releasing a software refers to the process of providing a created application to others for use. It is provided for every new release of the product build.

Bug fix refactoring is the process of improving the design of existing code by changing its internal structure which tends to improve software quality by improving design, improving readability, and reducing bugs.

Test plans are intended to plan for testing throughout software development life cycle. It should know the limitations and constraints of testing. In order to perform the stability part of the test, the user must identify and outline the basic kinds of data that are processed by the product without failing or causing failure.

Writing a capture-once, replay-on-all testing framework seems like addressing the problem, even if the experiences with this type of tests in desktop applications are not always positive, contrary to the desktop, mobile applications, such as CPU, memory usage, etc [6].

verification mechanism and, the developers are unable to record test scenarios all scripts are written prior to testing.

7. METHODOLOGY

mobile applications are much simpler, so test scenarios should retain manageable size. Unfortunately, the J2ME environment does not offer any system-level support with respect to handling GUI events and any other events for that matter. The user replaces this required and missing functionality with an automatic preprocessing of the binary code of the tested program, a process generally known as byte code level instrumentation or code injection.

The testing framework is then build upon RobotME – for designing automated recording of the interaction with a mobile application, regardless of the aspects of implementation. A test script recorded once, during development, on an emulated device, should run identically on other devices. The test validation phase is executed both on real devices and emulators of mobile devices, effectively making the testing shorter, easier and repeatable.

8. TESTING AND IMPLEMENTATION

a) Testing

Testing is a process of checking whether the developed system is working according to the original objectives and requirements. Testing is a set of activities that can be planned in advance and conducted systematically. It is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the global will be successfully achieved. Testing is like processes where inputs are received and outputs are produced. In the proposed work the GUI testing is done to improve mobile software efficiency.

b) Implementation

i) Mobile application software- Mobile Application Software as application software using on mobile classifies into three large groups by its features (CP, SP, EM) as follows.

ii) Mobile CP (Contents Provider) application- As the application software with high market share, the entrance service of WAP(Wireless Application Protocol) of mobile phone company such as MagicN of KTF, Nate, CPs of mobile phone companies have develop and serviced.

iii) Mobile SP (Service Provider) application- This is the special service application operating on VM (Virtual Machine), and it has been developed for long-term service rather than short-term selling, and used by download with the mobile phone through CP contract with mobile phone companies.

iv) Mobile embedded application- As the application with porting and mobile phone merchandising; this has been developed as request of mobile phone companies based on Virtual Machine, or on C by API of mobile phone.

The size problem can be easily illustrated. Unlike a Command Line Interface system, a GUI has many operations that need to be tested. A very small program such as Microsoft WordPad has 325 possible GUI operations. In a large program, the number of operations can easily be an order of magnitude larger.

The second problem is the sequencing problem. Some functionality of the system may only be accomplishable by following some complex sequence of GUI events. For example, to open a file a user may have to click on the File Menu and then select the Open operation, and then use a dialog box to specify the file name, and then focus the application on the newly opened window. Obviously, increasing the number of possible operations increases the sequencing problem exponentially. This can become a serious issue when the tester is creating test cases manually.

c) Experiments and Evaluation

i) Test case generation in server client environment

In this module, we simulate the PDA for implementing the test case. Mobile application program is shorter than desktop application one in developing term and market penetration term. Screen is the place where users make the most use of and communication is activated lively in mobile application program. Server has been composed that is Network Class and Serialize Decoding part and test case, test analyzer.

ii) Validating address book

The address-book application is a simple web application that stores contact data. It uses a single entity class, Contact, that uses the Java API for JavaBeans Validation to validate the data stored in the persistent attributes of the entity, as described in Validating Persistent Fields and Properties.

The Contact entity uses the @NotNull, @Pattern, and @Past constraints on the persistent attributes.

The @NotNull constraint marks the attribute as a required field. The attribute must be set to a non-null value before the entity can be persisted or modified. Bean Validation will throw a validation error if the attribute is null when the entity is persisted or modified.

The @Pattern constraint defines a regular expression that the value of the attribute must match before the entity can be persisted or modified. This constraint has two different uses in address-book.

- The regular expression declared in the @Pattern annotation on the email field matches email address of the form *name@domain name.top level domain*, allowing only valid characters for email addresses. For example,username@example.com will pass validation, as will firstname.lastname@mail.example.com.However,firstname,lastname@example.com, which contains an illegal comma character in the local name, will fail validation.

- The mobilePhone and homePhone fields are annotated with a @Pattern constraint that defines a regular expression to match phone numbers of the form (*xxx*) *xxx*–*xxxx*.

iii) Validating Message

To validate message application, initially the number is checked, it must be a valid 10 digit number. The text area can contain alphabets, numbers or combination of both and must not exceed the limit of the text. Finally ensure whether the message has delivered correctly to the client.

iv) GUI testing for application compatibility

As GUI test method of mobile application, the most popular usage methods in the mobile application software enterprise are calculator. Here we optimize for effective GUI use.

v) Algorithm for optimizing mobile applications:

Step 1: First check the title bar is displayed as calculator or not.
Step 2: Check the digit 0 is displayed in edit box of calculator.
Step 3: Check the arithmetic operations are performed correctly or not. For ex: 2+3=5,2*6=12 etc....
Step 4: Check the sqrt button shows the square root of a number or not. For ex: 81 click on sqrt button output is 9.
Step 5: Check if you click on 1/x button it shows can't divided by zero or not.
Step 6: Check if you click on button then it clears the content in the edit box.
Step 7: Check if you click on backspace button. Then it deletes the content one by one from right to left.
Step 8: Check if you select view menu and select a menu item scientific, then it displays the scientific calculator or not.
Step 9: Check if you click on = symbol, then it gives the result or not.

9. RESULT

Test performance result compares transmitted data in Mobile with the data in test scenario. Performance result of scenario can be confirmed. If test fails, test is discontinued and displays failure point to tester from relevant part. Fig.3 shows the comparison of memory size of applications before and after optimization based on the total number of applications used.

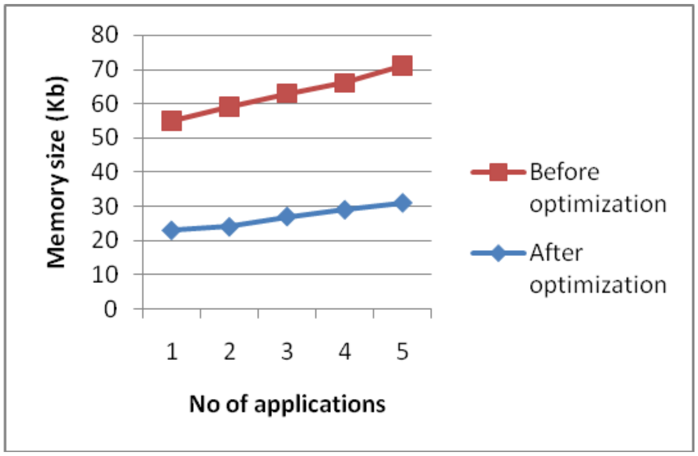

Fig.3 optimized value of mobile applications

10. CONCLUSION AND FUTURE WORK

Optimization offers cost-effective performance delivery in mobile applications. Mobile applications are executed on remote servers that render the output and send it back to the client. The calculator application optimizes the memory usage of the mobile software. The dynamic mechanism causes significant server load, while lots of applications for optimizing the performance. The optimization technique is applied for both client and server mobile application.

By optimizing mobile application the performance is increased and other mobile application aspects performance considerably increases.

Performance optimization is done at the time; the application is executed on the client machine. The dynamic optimizer lets the hardware cost remain low by providing the flexibility of customizing the optimizer for specific applications.

In future, mobile optimization can be used to defer many expensive network and hardware upgrades while reducing the cost of supporting the mobile usage. Developing applications for platforms such as Android require extensive testing as hardware configurations can greatly influence performance and optimizing for all mobile application features can be done.

11. REFERENCES

[1]Ahmad Saad Al-Mogren,*A Powerful Optimization Approach for the Multi Channel Dissemination Networks*, International Journal of Computer Networks and Communications, Vol 2, No 2, 2010, pp: 132- 140.

[2] Bettina Biel, Thomas Grill, Volker Gruhn,*Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application*, Journal of System and Software, Vol 83, No 11, 2010, 2031-2044.

[3] ChristosB.Anagnostopoulos, *Modeling and Architecture of Mobile Computing Systems*, International Journal of Wireless Personal Communications, Volume 42, Issue 3, 2007, 445-464.

[4] Emanuela G. Cartaxo, Francisco G. O. Neto, Patr´ıcia D. L. Machado,*Test Case generation by means of UML sequence diagram and labeled transition system*,2007, 1292-1297.

[5] Ioan Mihnea Iacob, Radu Constantinescu, *Testing: First Step towards Software Quality*, Journal of Applied Quantitative Methods, Volume 3, Issue 3, 2008, 241-253.

[6] Marinho F.G, Dantas, V.L.L, Andrade, R.M.C,*Testing requirements for mobile applications*, International Symposium on Control of Semiconductor Interfaces, Vol 1, No 7, 2009, pp: 555-560.

[7] Moumita Majumder, Sumit Dhar, *A Mobile Message Scheduling and Delivery System using m-Learning framework*, Journal of Telecommunications, Volume 1, Issue 2, 2010.

[8] Ren´ee C. Bryce, Sreedevi Sampath, Gokulanand Viswanath, Vani Kandimalla, A. Gunes¸ *Prioritizing User-session-based Test Cases for Web Applications Testing*, International Conference on Software Testing, Verification, and Validation, Koru,2008