Innovate QA/Test Processes with Model-Based Testing

Authored by Hani Achkar, DDI Health; Yaxiong Lin, TestOptimal **Quest 2012, Chicago**

A Superior Alternative to Traditional Software Testing Approaches

In today's highly competitive global market, software quality and the ability of development teams to quickly respond to changes are often the differentiating factors that set an organization apart from competitors. Model-based Testing (MBT) is an emergent, next generation software testing methodology, that enables QA and test engineers to revolutionize QA and testing processes to face today's challenges.

MBT has been widely used in Telecom, Automotive and other industries for testing embedded systems since the 1990's. This technology is now available for application software testing through numerous commercial tools which are at the forefront of this wave of technological advancement. MBT benefits include early detection of requirement defects, improved test coverage, shortened testing cycles, enhanced regression cycles and automated test generation. What we demonstrate in this article is based on practical experiences in applying MBT to mission-critical systems.

First Steps

The first step in the MBT process is to create a behavioral model of the Application Under Test (AUT) as derived from requirements and specifications. Typically in MBT the model is proposed in Finite State Machine (FSM) and Extended FSM (EFSM) representation, although other formats have been applied, including Petri Nets and UML State Charts. It is a typical experience that the modeling process in and of itself often reveals inconsistencies and deficiencies in the requirements being addressed and as such is a highly valuable formal validation step and effective "testing" stage.

Models can easily give rise to test cases which can be visualized graphically providing an intuitive and effective communication and documentation mechanism. Extending this with intuitive animation of model execution by employing MBT software tools enables engineers to understand the dynamic behavior of an AUT before it is even implemented. As a result, requirement defects associated with dynamic interaction may be found which otherwise may only be discovered during acceptance testing or later test stages.

Effective Test Case Generation

Traditional, non-MBT test case design depends heavily on the ingenuity of individual designers. This results in a vast disparity of quality in test cases and test coverage between designers. A good MBT tool can provide a framework that enables QA professionals to achieve better consistency and quality in test design with much higher system coverage than traditional means can provide for.

There are several methods for generating test cases/sequences from a model, including random walk, shortest paths, Chinese postman walk and others to achieve desired coverage criteria or goals. Because the models describe the behaviour of the AUT, the traversal of paths from the model represents a series of steps that effectively exercise or test the AUT.

Efficient Handling of Changes

In today's agile development processes, the ability to respond to requirement changes is imperative. Traditional manual test case creation is inefficient, labor intensive and sluggish in responsiveness. As such it is avoided in agile development and is replaced by exploratory testing and Test Driven Development (a form of automated unit testing sufficing for automated testing), foregoing the benefits offered by formal system testing.

With an MBT tool this limitation can be removed. While development teams are busy implementing an application, QA/test teams can build and validate models in tandem. Changes are easily handled through adjustments to models that automatically generate/regenerate test cases or test vectors for automatic execution against the target AUT. What may have taken days now only takes minutes, in some instances, to achieve with an MBT tool.

Realistic Load / Stress Testing

Models developed for system testing can be easily re-purposed for load/stress testing. Models of AUT behavior can be executed concurrently simulating hundreds or thousands of virtual users utilizing the AUT in realistic scenarios. These generated loads are not random or a predefined selection of arbitrary http packets/requests. Instead they accurately simulate populations of users running legitimate scenarios supported by the AUT requirements. Such realistic simulation of production load would be very difficult to achieve with non-MBT based load/stress testing tools.

Practical Experiences

DDI Health (DDI) is a leading e-health service provider in Australia, offering IT services, PACS for Radiology and clinical IT solutions across the diagnostic and general health care sector. As with any service provider, the quality of DDI's software is vitally important to the success of its business. DDI has implemented MBT universally across its entire software offering. As a result, substantial improvements in defect detection during modelling and system testing have been evidenced by dramatically reduced numbers of defects found in User Acceptance Testing and production.

TestOptimal has empowered DDI with the ability to automate the functional testing of a large web based medical results portal achieving over 90% requirements covered by model based/derived automated tests improving release turnaround times by over 60%. Load testing was achieved through repurposed models at minimal cost and minimal time.

Conclusion

MBT is an efficient test design approach and an effective test automation methodology that has been proven at DDI and many other organizations. With the help of an integrated TestOptimal MBT solution, DDI has drastically reduced defects found after software release, improved test coverage and shortened test cycles. What has been accomplished at DDI is astounding. We encourage all QA and test engineers to consider MBT and experience what MBT can do to supercharge the QA and testing process like no other test automation tool can.