TEST HARNESS

Project 2

Version 1.0
5th October 2016

CSE 681 – Software Modeling and Analysis
Instructor: Jim Fawcett

Sahil Surendra Gupta
SUID: 74146-3736
sagupta@syr.edu

The Test Harness developed fulfills all the functional requirements stated for project 2. It implements most of the concept stated in the Operational Concept Document and fulfills all the stated obligations. Apart from the scheduler, all functionalities described in the OCD are implemented.

Following is the list of features/concept that that the implementation covers

1. Provide well defined structure for test request – The XML file describes structure of test request
2. Accept test requests from client in XML format – Test Harness allows the client to submit multiple test requests for processing through its '/t' switch on the command line interface.
3. Process each test request in an AppDomain - Test Harness creates one AppDomain for every test request and processes all tests for that test request in the created AppDomain.
4. Provide a queue for serializing multiple test requests – A queue allows en-queuing and de-queuing of test requests.
5. Log results of each test execution – Logs of every test are saved with author and current date time stamp in a file.
6. Support client querying of past test requests – Clients can get details of individual test, details of all his past tests and a summary of all tests executed by the test harness.

Following are the list of functionality not addressed in current implementation

1. Scheduler component is not implemented as it is a low priority feature. Even without the scheduler, the test harness is fully functional and users can simulate the scheduler means of scripting.
2. Logger for debugging test harness is not implemented due to limited time constraints. Again, even without the logger, no functionality of the test harness from a user's perspective is affected.
3. A database management system could not be implemented. To fulfill the concept, a file based database has been implemented which provides the same functionality with the disadvantage of being less efficient and organized.

Weighing the importance of implemented features, we can see that the original concept was indeed a practical design and could be implemented in the given time frame. Only a handful of non-critical features were not implemented.

There are however certain details that although provide the functionality described the OCD, are not structured the way it was proposed. These are,

1. Due to the unavailability of a DBMS, the implementation of the DatabaseManager with ReportGenerator could not be separated. In current implementation, both these packages are implemented as FileManager that does the work of storing test data and also displaying it in proper format when requested by the client. With the availability of DMBS, we can add more extensive support for querying and report generation.
2. The RequestQueue package was not needed because of the availability of .NET framework class libraries. The existing Queue implementation form the Systems.Collections.Generic class could be used.