

Automation of Testing in the Distributed Common Ground System - Army

Michael Eng, TJHSST 2009

Mentor: Mark Pumphrey

Northrop Grumman Mission Systems Sector

Purpose and Scope of Project

The purpose of this project is to produce a proof-of-concept prototype system to automate the time-consuming and laborious task of testing large Java language software systems. Because DCGS-A is actively being used by the United States government, revisions and improvements to the DCGS-A system need to be quickly and efficiently tested to mitigate defects and security flaws. I will be using testing through decimation in which if each method is thoroughly tested then the whole system will be deemed as tested.

Once a testing suite is written, a means of running it on a regular schedule will need to be determined. This test scheduler will then help maintain the system as a whole and to increase efficiency. Tests can be run overnight and will produce results that will be available at the opening time of business. Running a testing suite overnight will also improve performance by offloading tests after hours. As part of the system testing, security aspects of the software will be examined and tests will be designed to exploit potential security weaknesses.

Background

Throughout history, armies have known that one of the most essential keys to victory is intelligence. Without knowledge of your enemy or of the battlefield, your effectiveness is reduced exponentially - defeat is virtually assured. In the Civil War, scouts and spies were used to collect intelligence on key factors like terrain and enemy movements. More recently, militaries have bolstered their intelligence-gathering arsenals with unmanned aerial vehicles (UAVs), satellites, and ground reconnaissance systems. However, up until a couple of years ago, the large assortment of intelligence systems had always been collected and analyzed separately, which increased the chances of miscommunications due to problems such as latency.

DCGS-A is an emerging intelligence system that uses data fusion to consolidate and correlate intelligence from a variety of systems ranging from signal, electronic and imagery to human intelligence and weather sensors. DCGS-A's main purpose is to provide a cohesive and common intelligence database to commanders so they can analyze the input from many different intelligence sources. Because of the tactical nature of DCGS-A most of the system is mobile and transported on vehicles and typically is set up in Tactical Operation Centers on the battlefield. Other parts of the system are designed to be fixed, and are located in the rear battle area such as a theater's regional operations center or a base back in the U.S. Finally, sensors are placed on the ground and embedded in aircraft to maximize range and effectiveness.

DCGS-A receives and processes data from multiple systems into one common database. These systems consist of the following:

- *All Source Analysis System-Light (ASAS-L)
- *Analysis and Control team-Enclave (ACT-E)
- *Block II Analysis and Control Element (ACE)
- *Common Ground Station (CGS)
- *Counter- and Human-Intelligence Management System (CHIMS)
- *Digital Topographic Support System-Light (DTSS-L)
- *Enhanced Trackwolf
- *Ground Control System (GCS)
- *Guardrail/Guardrail Information Node (GRIFN)
- *Integrated Meteorological and Environmental Terrain System Light (IMETS-L)
- *Prophet Control
- *Tactical Exploitation System (TES)

(<http://www.gdc4s.com>)

The system also contains an integrated application framework called the Multifunctional Workstation (MFWS) that allows developers and analysts to rapidly add new tools into the system. DCGS-A replaces nine programs of record (PORs) that were previously used by the Army as intelligence sources during conflicts. [Pollachek 2008]

Procedure:

My project would mainly consist of two parts. First, I will work on automation of testing of certain DCGS-A classes by using JUnit, a testing suite used to verify the correctness of methods and classes. Secondly, I intend to implement a testing schedule so that tests can be run in bulk overnight to increase efficiency. I will be coding and executing the JUnit tests in the Eclipse platform.

Software

Language: Java 1.5.0_11

Nonstandard packages: JUnit.jar

Development environment: Windows XP SP2, Eclipse- Workshop For WebLogic, Ant 1.7, JBoss 4.0.5GA

Testing and Analysis

Programs will be deemed successful if they successfully and accurately test the methods and classes that they are assigned to test. This can be verified by comparing results of the tests that I wrote to results of tests for the same classes/methods written by other people.

Expected Results

I expect to create multiple tests throughout this mentorship period for multiple classes within DCGS-A. I also expect that if these tests work successfully, then I can implement a program that will schedule test runs of programs overnight, such that when designers arrive in the morning, they will receive results of said tests and be able to identify bugs and flaws to fix, thereby increasing productivity and saving time and resources normally used for testing.