SESP 2010

Session:	Session: Test Procedures (21)
Type:	Concurrent Session
Date:	Thursday, September 30, 2010
Time:	11:00 - 12:00
Room:	Einstein
Chair:	
Co-chair:	
Remarks:	

Seq	Time	Title	Abs No
1	11:00	An Integrated Development Environment for Spacecraft Checkout Test Procedure Development Vasanthakumari, U.N. ISRO SATELLITE CENTRE, INDIA Spacecraft Testing during the AIT Phase in an involved activity. Ensuring Automation, repetivity of the tests during various phases of AIT are quite complex in nature. To prepare the satellite test procedures in advance in an English like language with features like automated checking of the related parameters, pre-requisites, linking of test procedures are available in the CHECKOUT COMMAND LANGUAGE (CCL) designed and developed in-house at ISRO Satellite Centre, Spacecraft Checkout Group. The integrated development environment for the preparation, validation, updation and maintenance of the Test Procedures helps in a professional way of managing the satellite test procedures. It maintains a set of libraries, directories based on the sub-system wise, Phase wise etc. which helps in choosing the appropriate test procedure files and creating the Test Schedules. This paper discusses the various features available in the IDE for Spacecraft Test Procedures designed and developed with the idea of helping the test engineer during the testing phase. It provides ease of operation, quick way of generating schedules etc. Simulation feature added to this IDE provides the test engineer in evaluating the test time required, checking of conflicting procedures and also helps in training the new test engineers.	
2	11:30	Working Environment for Test and Simulation (AWETS) Kolkman, H.[1]; Timmermans, L.[2]; Neefs, M.[3] [1]TASK24, NETHERLANDS; [2]NLR, NETHERLANDS; [3]Dutch Space, NETHERLANDS Current simulation tools provide limited support for controlled execution of the work, as required for most space related projects. In preparation of a test or simulation, the user needs to define and/or setup his/her own working environment. This entails related activities such as definition of monitoring and control data, test definition, test execution and test analysis and reporting. These preparatory and supporting activities lead to additional costs for projects, often not accounted for. The Automated Working Environment for Test and Simulation (AWETS) project is a development to improve the efficiency of test and simulation activities, in particular for EuroSim based systems. The project was funded by the Netherlands Space Office (NSO) and executed by a consortium of NLR, Dutch Space and TASK24.	

AWETS focuses on the "Facility Monitoring & Control" part of a test/simulation system as defined in ECSS-E-10, Space Engineering - System Modelling and Simulation. AWETS monitors and controls test and simulation execution. AWETS also adds tooling to efficiently prepare a test and simulation session beforehand and analyze the results afterwards.

The main objectives of the AWETS system are:

- Create a re-usable test automation platform that can easily be expanded for specific project needs and test subsystem interfaces.
- Automate simulation and verification activities during the complete spacecraft lifecycle.
- Integrate Test Data Definition, Test (Script) Definition, Test Execution and Test Results Analysis in one environment.

Additional to these main objectives there were objectives such as to keep the platform portable and light weight (it should run on standard Linux and Windows PC) and to use open source and open standards as much as possible. Also the system should provide detailed logging and high level test result summaries including system requirements coverage.

The open source Eclipse framework was selected as the base for the AWETS platform because of its high degree of acceptation in the development community, extensive library support and proven flexibility to build dedicated user applications using the plugin architecture. The available Eclipse SDK's also provide excellent environments for development and debugging of test scripts.

Java was selected as the language for the tool implementation because of its platform independence, the availability of extensive open source libraries (e.g. for graphics and XML data binding) and because it is the Eclipse native language.

The AWETS Test Database is implemented in XSD/XML and basically stores Test Specifications which further split into Test Cases and single Test Steps which can all link to System Requirements. The XML solution is light weight and via the Eclipse Modelling Framework (EMF) dedicated XML editors are generated from the XSD which plugin into the Eclipse framework.

The Test Steps in the XML database refer to single Test Script files. Test Scripts have to be written in Java though the choice for Eclipse and the AWETS architecture supports implementation of other test script languages in future. Java offers the advantage of AWETS being capable to use so-called dynamic compilation (compiling and loading the test script code at test execution time) which provides great test script development, editing and debugging facilities in combination with the Eclipse Java SDK.

For Monitoring & Control (M&C) data definition it was decided to use XML and comply with the XML Telemetric and Command Exchange (XTCE) standard. XTCE aims to guarantee the interchangeability of M&C specifications between multiple partners in a space project. It allows development of standard tooling such as automatic code generation from M&C specifications for data packet processing as was demonstrated successfully in AWETS.

AWETS provides basic user interface capabilities to monitor test subsystem data (numerically or graphically). For more sophisticated data visualization, a coupling with NLR's Vincent tool was created via a built-in data server. Vincent provides a tool suite for rapid development (and evaluation) of simple to highly complex Synoptic Displays and

Human Machine Interfaces.

Finally, the above mentioned tools and design concepts were successfully implemented in an AWETS demonstrator system which drives a simple simulation model running in EuroSim.

The presentation on AWETS will focus on the design drivers, the architectural design and the results and findings of the AWETS demonstrator system.