System of Systems Automated Testing in the Ground Segment Reference Facility

Jean-Christophe BERTON, Eduardo GOMEZ[(1)]
Louis HANNA, Mario PINTO [(2)]

[(1)]European Space Operations Centre
Robert-Bosch-Strasse 5, 64293 Darmstadt, Germany

[(2]ETAMAX Space GmbH
Robert-Bosch-Strasse 7, 64293 Darmstadt, Germany

INTRODUCTION

Since 2011, ESA has been focusing more and more on End-to-End (E2E) Integration and Testing (I&T) of the Mission Operations Infrastructure (MOI) products. Therefore,the European Space Operations Centre (ESOC) established the End-to-End Ground Segment Reference Facility (GSRF) as the one-stop environment where the system level I&T of the MOI takes place, as in [1].

One of the major I&T activities performed in the GSRF relates to the automation of the MOI systems testing. In this domain, ESOC already developed a number of tools. Due to the increased complexity of current distributed systems, the development of flexible validation tools and strategies has become more and more relevant in the last years. Although a number of tools exist in the market for test management and automation, none proved to be designed for covering the full end-to-end validation for highly specialised and distributed ground segment components.

In order to gain in efficiency and reliability in the domain of automated I&T, the GSRF enhanced the automation, monitoring, control and test performance of the facility as part of the provision of a prototype environment covering end-to end test Ground Segment validation in a modular way. This allows "plug-and play" type testing, where a specific software module can be tested without having the full system available or without the need of developing a dedicated test framework simulating the full system.

This prototype environment, called “E2EVAL” results from the GSTP funded “Building blocks for End-to-End validation and management of distributed ground segment systems components”activity and rationalises the approach to test automation of large system assemblies that the identification, investigation and reproduction of failures at early stages of ground systems developments, during their development lifecycle or use in operations. Performing system level testing in operations representative test assemblies, reduces effort and increases the effectiveness and reproducibility of the testing.

BACKGROUND

The E2EVAL activity builds on top of predecessor study results like the Automated Regression Testing (ART), Man-Machine Interface Testing (MMIT) and SMURF, as in [2], which made it possible to test different software components in early stage of the mission phases automatically. They were implemented to record and simulate any user activities to interact with the System Under Test (SUT). The overall test automation architecture developed during the previous project MMIT2 is illustrated in the Fig. 1 below.

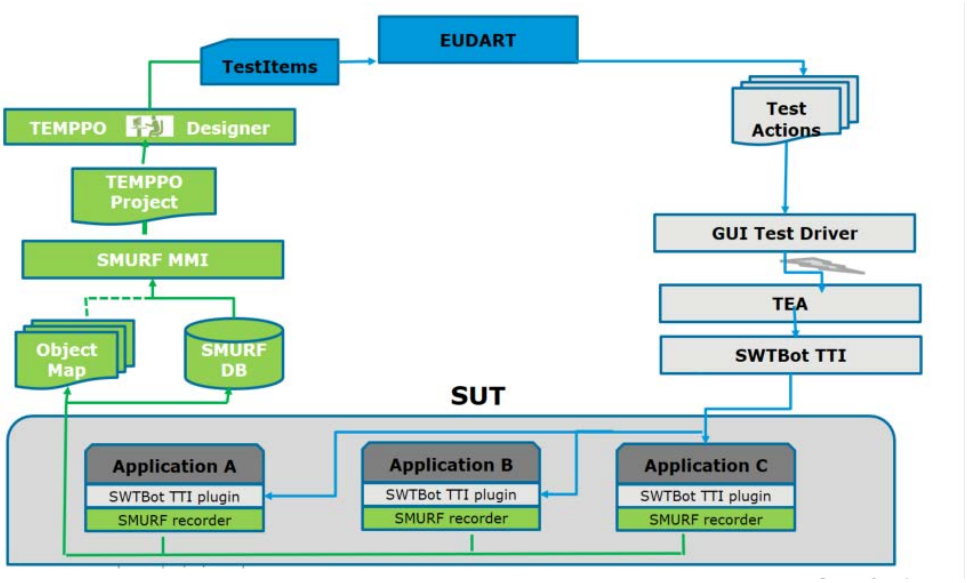


Fig. 1. The MMIT architecture overview

The MMIT concept of test automation for any subsystems consists of two mainstream lines: The left part of the diagram shows how the tests are designed and automated (this includes collecting data to support the automatic recognition of controls), the right part of the diagram shows how the test are executed . The test execution tool EUDART is used to read automated test plans, execute them and collect the results. The whole automation relies on the so called "object maps" that identify each control by its specific characteristics such as the java class implementing it or its location within the hierarchy.

Genuine User Interface Test Driver (GTD)

The Genuine User Interface (GUI) Test Driver receives test actions requests and routes them to the local agent on a (possibly remote) machine. Test actions are standardised.

Test Environment Agent (TEA)

The TEA should be running on the target machines where test steps have to be executed. It translates the generic test actions into a specific command to the underlying technology (currently based on SWTBOT).

SWTBot Test Tool Interface (TTI)

The TTI is based on the open source packet SWTBot. It activates actions on the controls. It has been enhanced and some new actions have been introduced as detailed further below.

SYSTEM DESIGN OVERVIEW

The knowledge gained during the previous activities has showed the need for a tool that increases efficiency of the testing and offers a shared and simplified interface that hides the complexities of setting up the test environment to the user. This tool is a Web Based Application, accessible from any browser inside the ESOC network. It provides an analysis of the test results in three different levels of details, allowing any operator to quickly identify the overall status of the results and analyse any detected failure.

The interaction between the different modules of the framework as well as their interfaces to the outside is depicted in Fig. 2. The Catalogue Management Module deals with the "cataloguing" and listing of all relevant information to the execution of test schedules. The Report Management Module takes care of the parsing of reports (originated from the execution of test schedules) providing different presentation layers to the user based on the outcomes of the Result Evaluation Module, which is there to to automate the analysis of the reports by identifying regressions and known problems. The Execution Management Module orchestrates the execution of test schedules by the

user, on target machines deployed with the target chain versions. This module heavily relies on Jenkins to do the actual work of execution.

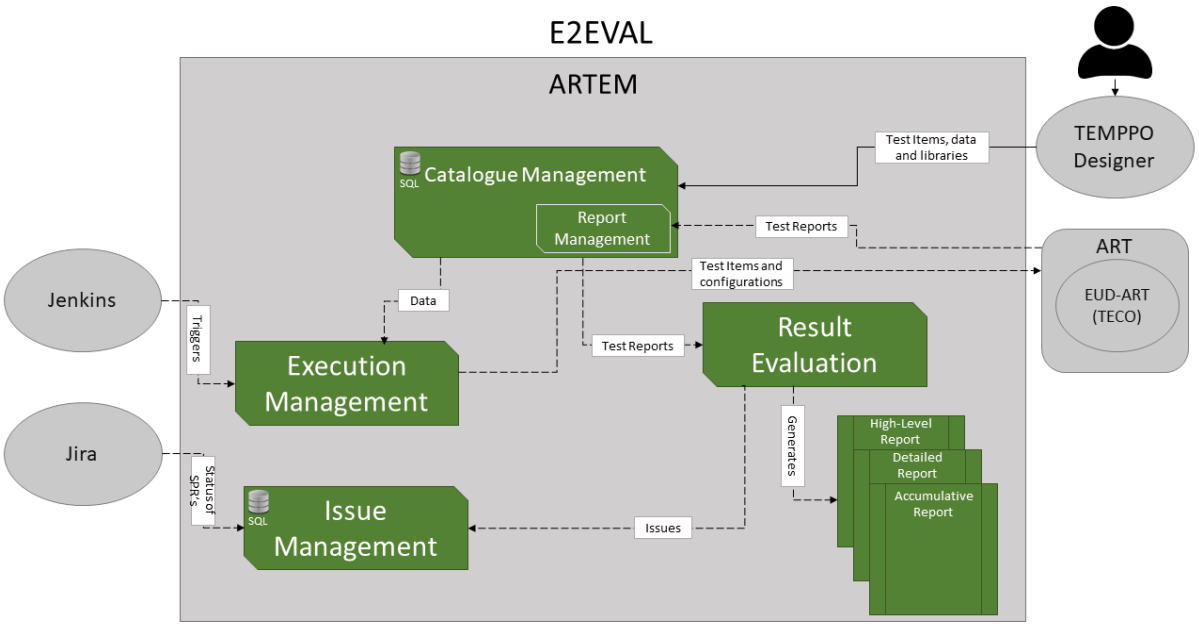


Fig.2 The E2EVAL Architecture

Users accessing the E2Eval framework can do so in one of three distinct roles. The role with the lowest set of privileges grants the user read rights. (e.g. to consult the reports of the test executions, to consult the catalogue of existing test schedules, chains etc.). A second role adds execute permissions on top of the previous role. Here the user is also allowed to trigger the execution of test schedules for a selected chain and target system. The last role, adds write permissions to the previous roles. This type of user can also, through the E2EVAL framework, insert new data (for example a new mission, a new chain or chain version, etc.).

The framework can be integrated with different environments. It is used to integrate and test any infrastructure or mission specific products. Some of the data is generated from an external tool and then fed into the E2EVAL via interface module, which let them stored properly.

Catalogue Management Module CMM

The CMM is responsible for storage all required data of the tests in a centralised database. It contains the test reports as well, that have generated after finishing of the executions of the Integration and Testing.

The majority of the information is stored in the CMM in a structured manner, which requires an intermediate layer to interface between the others modules. All the data stored in the module can be displayed in a general view or in a specific structured view or in a free adjusted view of the required tables.

The information is to be added through another insertion module either manually or automatically. The insertion of the data should be done before calling the executions platform of the framework.

The catalogue lists in a single central repository all the individual products that can be tested, either standalone or in test assemblies. The catalogue therfore also provides the capability to list all different test assemblies and related test chains that details which individual products builds up with which specific test chain. The catalogue associates all the available test schedules and can generate the relevant test project depending on the mission configuration, chain version and target environment details to generate them from the relevant test schedules. All individual components of the catalogue and their relations can either be inserted through the UI of the web-application or

directly from the database, providing also export functions to generate the tables as an excel file. The CMM view also offers filtering capabilities based on the column values.

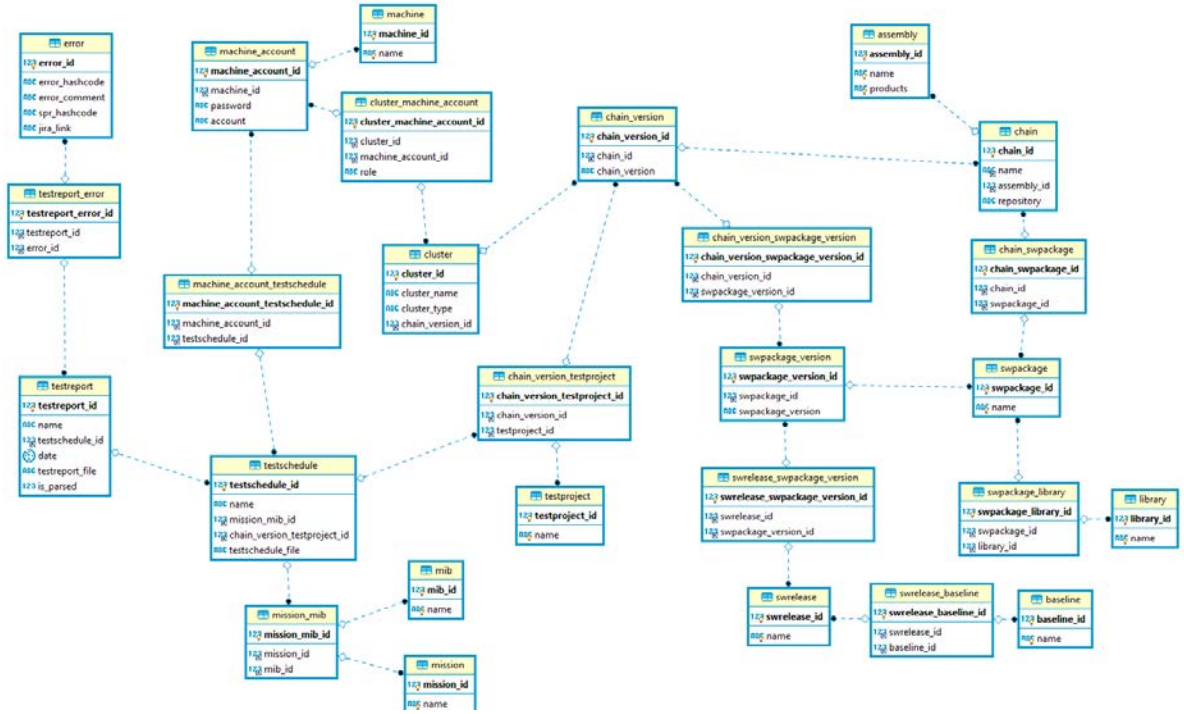


Fig. 3 The E2EVAL database model

Execution Management Module EMM

This module contains an overview about the systems that will be used in the executions phase of the testing. It will trigger some specific Jenkins job to start the testing on the target machine. All the selected test items in the module will be having the availability to be deployed in this test run or not. The configurations of the machines can be chosen from this module to avoid any misleading while using the scripts.

The execution module allows the user to select a cluster of machines and trigger the execution of test schedules and the preparation of the environment to run these test schedules either on one single standalone machine or on a cluster of a set of machines. The test schedules can therefore be user-defined and selected from the catalogue. The framework cooperates with dedicated Jenkins jobs to perform the execution of the test schedules and the preparation of the environment

Result Evaluation Module REM

The REM module provides the users with the test results in a comparison view with different test sets of either the same target machine or different missions. The test reports are the output of the test execution tool EUDART in the format of XML file. The test engineers can easily examine and evaluate the regression tests of the schedules.

The reporting module permits the upload of already existing reports through the UI and offers the user with the capability to see:

- a parsed version of the test reports in three different levels of readability
- a high-level report with only the results of the high-level test cases and information about the SUT and execution time
- a detailed report with all the levels of the report and their success state and additional information
- an accumulative report that is user-defined and compares the execution of different test reports

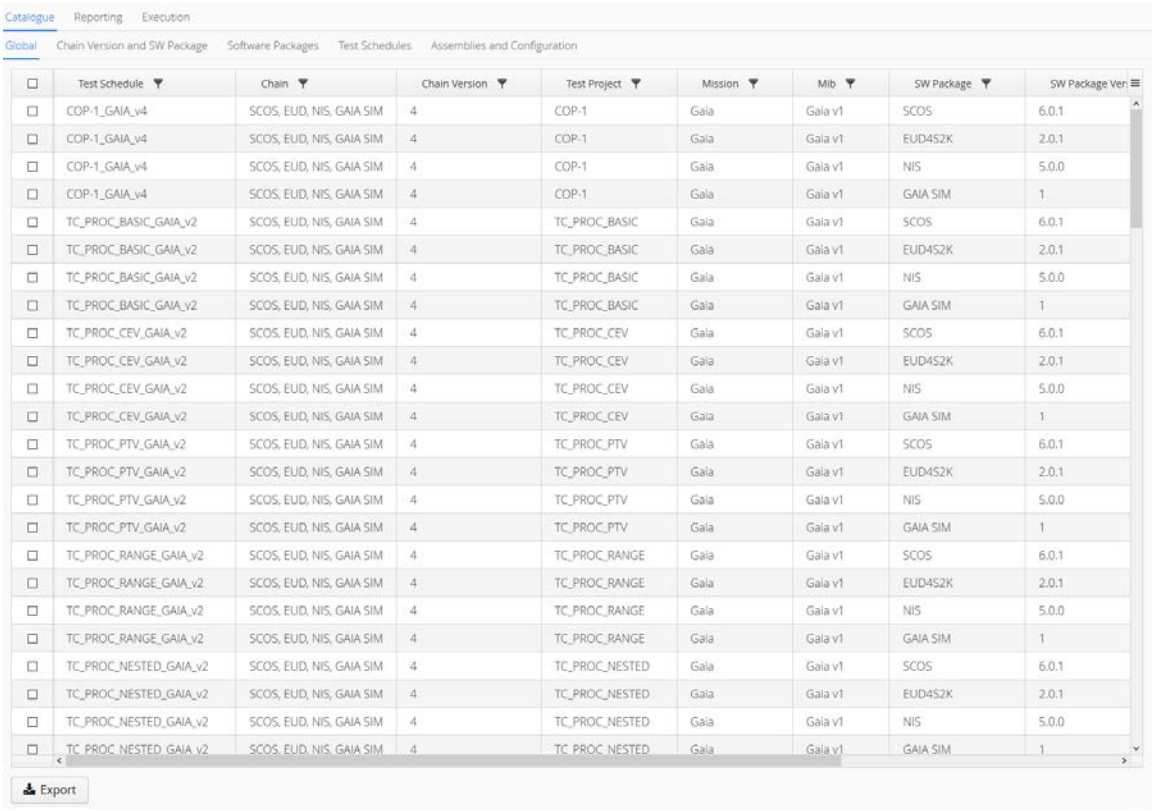
The user can also trigger a view in the accumulative report to only see the regressions, perform an automatic detection of errors in reports, jump to the first error of a report and export a test report as an export document.

Issue Management Module IMM

This module has the functionality to insert the new software problem found during the test or to retrieve the old software problem, which are stored in the database. It gives the tester the opportunities to update and validate the current problem reported on the software

E2EVAL IMPLEMENTATION

The E2EVAL framework provides direct access to the database content, as in Fig. 4, where pointers to test items, test reports, and the information that identifies these files are stored. The Test Schedules are generated through the TEMPPO Designer tool, and a pointer to them is stored into the Database. The data that defines a Test Schedule, is inserted into the Catalogue through the UI and ready to be accessed by any user. It ranges from the configuration used, to the individual products that it requires, to the machines where it can be executed.



Test Schedule	Chain	Chain Version	Test Project	Mission	Mib	SW Package	SW Package Ver
COP-1_GAIA_v4	SCOS, EUD, NIS, GAIA SIM	4	COP-1	Gaia	Gaia v1	SCOS	6.0.1
COP-1_GAIA_v4	SCOS, EUD, NIS, GAIA SIM	4	COP-1	Gaia	Gaia v1	EUD4S2K	2.0.1
COP-1_GAIA_v4	SCOS, EUD, NIS, GAIA SIM	4	COP-1	Gaia	Gaia v1	NIS	5.0.0
COP-1_GAIA_v4	SCOS, EUD, NIS, GAIA SIM	4	COP-1	Gaia	Gaia v1	GAIA SIM	1
TC_PROC_BASIC_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_BASIC	Gaia	Gaia v1	SCOS	6.0.1
TC_PROC_BASIC_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_BASIC	Gaia	Gaia v1	EUD4S2K	2.0.1
TC_PROC_BASIC_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_BASIC	Gaia	Gaia v1	NIS	5.0.0
TC_PROC_BASIC_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_BASIC	Gaia	Gaia v1	GAIA SIM	1
TC_PROC_CEV_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_CEV	Gaia	Gaia v1	SCOS	6.0.1
TC_PROC_CEV_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_CEV	Gaia	Gaia v1	EUD4S2K	2.0.1
TC_PROC_CEV_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_CEV	Gaia	Gaia v1	NIS	5.0.0
TC_PROC_CEV_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_CEV	Gaia	Gaia v1	GAIA SIM	1
TC_PROC_PTV_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_PTV	Gaia	Gaia v1	SCOS	6.0.1
TC_PROC_PTV_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_PTV	Gaia	Gaia v1	EUD4S2K	2.0.1
TC_PROC_PTV_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_PTV	Gaia	Gaia v1	NIS	5.0.0
TC_PROC_PTV_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_PTV	Gaia	Gaia v1	GAIA SIM	1
TC_PROC_RANGE_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_RANGE	Gaia	Gaia v1	SCOS	6.0.1
TC_PROC_RANGE_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_RANGE	Gaia	Gaia v1	EUD4S2K	2.0.1
TC_PROC_RANGE_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_RANGE	Gaia	Gaia v1	NIS	5.0.0
TC_PROC_RANGE_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_RANGE	Gaia	Gaia v1	GAIA SIM	1
TC_PROC_NESTED_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_NESTED	Gaia	Gaia v1	SCOS	6.0.1
TC_PROC_NESTED_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_NESTED	Gaia	Gaia v1	EUD4S2K	2.0.1
TC_PROC_NESTED_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_NESTED	Gaia	Gaia v1	NIS	5.0.0
TC_PROC_NESTED_GAIA_v2	SCOS, EUD, NIS, GAIA SIM	4	TC_PROC_NESTED	Gaia	Gaia v1	GAIA SIM	1

Fig. 4 Global Catalogue View

In a separate view of the Web-application, the Test Schedules present in the Catalogue can be selected and executed, given the machine/account constraints that each has. The execution is supported by a Jenkins job that commands the machine to execute the Test Item using the ART/MMIT framework. As a result, a EUDART report file is provided (by the Software product EUDART) and stored in the Database.

The Reporting view of the Web-application can be accessed to view all Test Reports executed to date and a total of three different ways to analyse each of them, from a High-Level perspective to a detailed one. A dynamic grouping of several different runs is also possible.

Finally, the parsing, performed by the Reporting module, allows the detection of issues that resulted in the failure of Test Cases. These issues are handled by the IMM that stores every new and unique issue in the database. Interface-wise, this allows the tracking of issues and cross comparison between reports. This module is connected to a Jira Software SPR Module, in order to provide even more relevant information and direct access to the reported problem reports.

All the tables displayed in the UI of the Catalogue Management Module can be filtered and exported as an excel file. The export takes the current table, including all the customizations that were performed in the UI and produces an excel file with a table visually similar to the one in the UI of the application.

Report Management Module

The RMM gives a high-level view over the available reports, ordered by date as in Fig. 5. A more detailed view of the report can be accessed by selecting the relevant test report result.

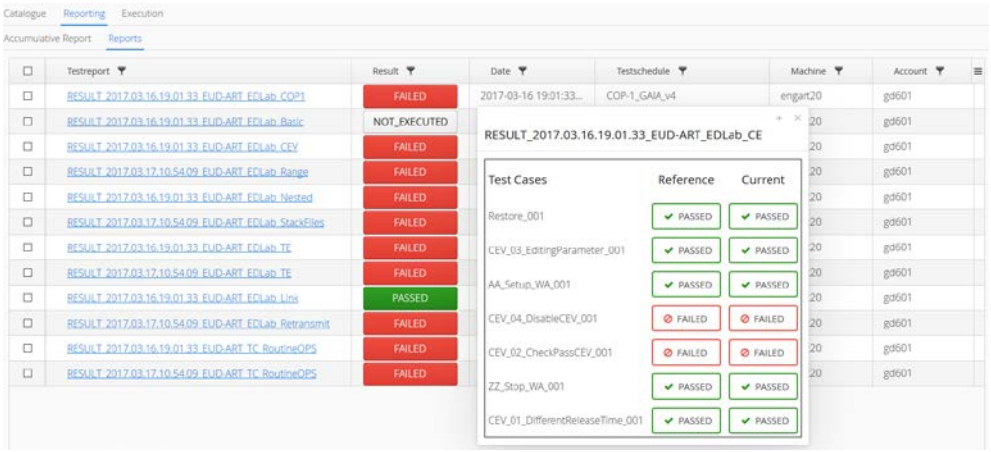


Fig. 5 High-Level Report View

To obtain an Accumulative Report, the user has to select one Chain Version, and from the available Test Projects, one or more of them must also be selected. The Software Package Version selection is optional. Finally, the last step required is setting the desired time frame to obtain the popup view aas in Fig. 6 with the Accumulative Report where the user can hover the mouse on the red cells of the grid and a detailed report of the errors will be presented. The user can the compare test results of different software package versions in a similar test assembly, using the same configuration and along the same test cases to highlight any regression or sporadic behaviour.

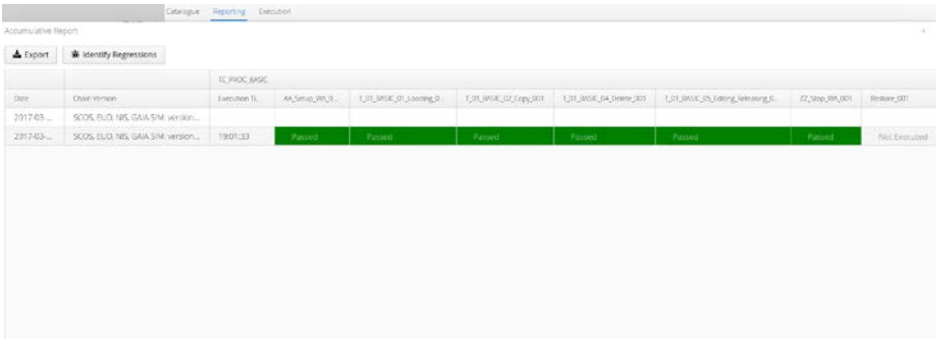


Fig. 6 Accumulative Reports Results View

The user can also access the Detailed Results of each step of the Test Report execution through a popup view with all the executed steps. In addition, the user can then click on the ‘Go to falling Step’ button for further information about the step that caused the failure as in Fig. 7.

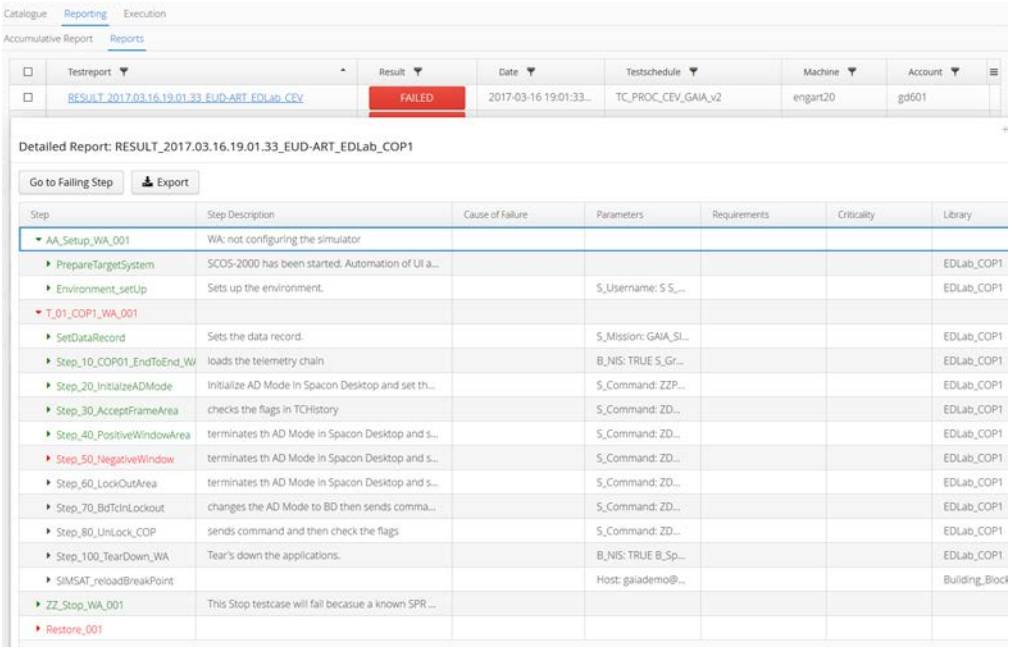


Fig. 7 Detailed Report View

Test Execution Management Module

With the purpose of executing Test Items, the EMM contains an external interface with the Jenkins server hosted by ESOC. Every time an execution of a Test Item is requested, an existing Jenkins job made for that purpose is triggered which performs all the necessary steps to execute the desired action. Two actions are possible: deployment of the E2EVAL plug-ins and actual test execution.

The Deployment tab allows the user to request the deployment of the necessary ART/MMIT software to run the automated execution of tests, as shown in Fig. 8 by selecting the target Cluster.

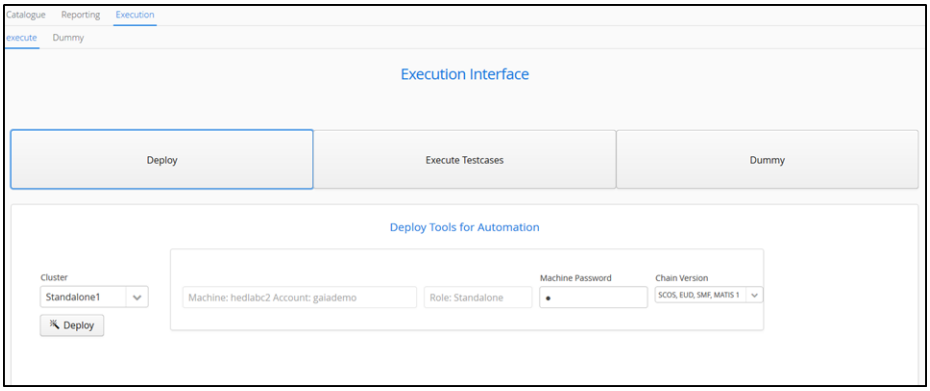


Fig. 8 Deploy necessary tools for test automation on a cluster

If the Cluster of machines is already equipped with the Art?MMIT tool, the user can request the execution of test schedules on this cluster using the UI shown in Fig. 9, where the targeted chain version and the available test schedules have to be selected before being executed by triggering the execution process.

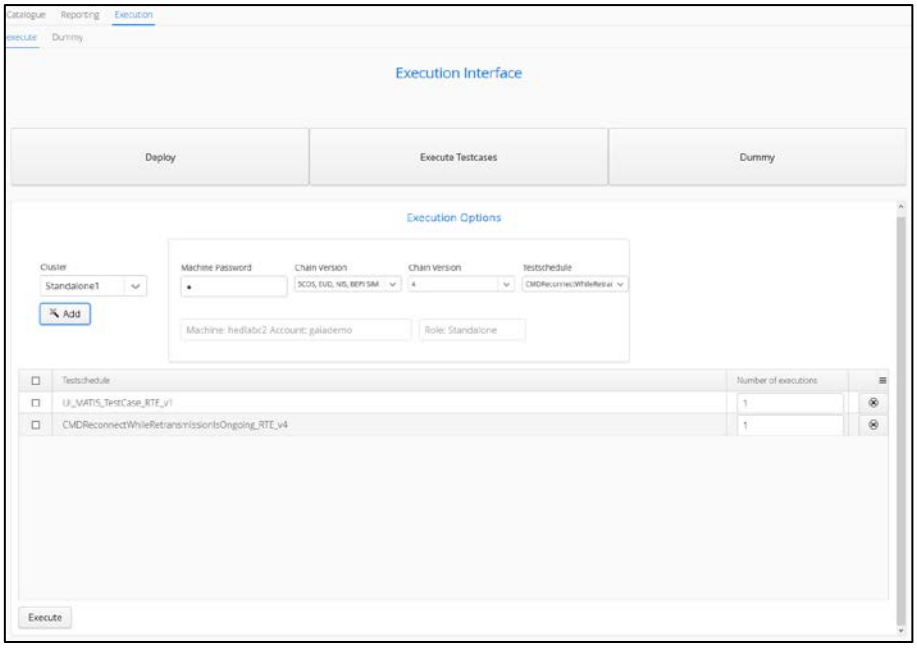


Fig. 9 Execution of test schedules on a cluster

An Administrator User has additional privileges to access to the 'Developer tools', offering the possibility to edit and add content to the application's database: new test schedules, new test reports, new clusters, new configurations, new software packages and versions as well as new test projects, test assemblies and chains.

CONCLUSION

The following goals have been reached during the course of the activity:

- Implementation of a web interface to capture the relevant information needed to validate a ground segment software chain from an end-to-end perspective
- Development of a Catalogue Module to store and display all the relevant infor-mation and their relations like: chains, assembly, testproject, machines, accounts, testreports, missions, mibs, and others. This module answers questions like "what can currently be tested" or "what is compatible with…".
- Development of a Report Management Module capable of parsing a test report and generate three different types of reports out of it. (High-Level, Detailed and Accumulative).
- Development of a Result Evaluation Module that works in collaboration with the Report Management Module to help identify regressions between executions of test schedules and to detect errors that might have occurred previously facilitating the work of the test engineer.
- Development of an Execution Management Module that provides the user the pos-sibility to select test schedules that he wants to execute on a specific cluster of machines with a compatible chain version deployed and request the execution. Several Functional test cases were successfully automated.

Although the implementation of these goals can be seen in the framework itself, not all functionality have been reached. In particular, the extension to test web-based applications as well as the adaptation to European Ground Segment Core Component (EGS-CC) based systems.

REFERENCES

[1] J.C. Berton, E. Gómez, C. Smith L. Teixeira and K. Widegård, "The ESOC End-to-End Ground Segment Reference Facility," in AIAA Spaceops, Marseille 2018

[2] E. Gómez, B. Semke and S. Mohacsi, "Enhancing Test Automation of Ground Data Systems through Direct Access to the User Interfaces," in AAIA Spaceops, Daejeon 2016.