ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM001

EXTENDING THE COVERAGE OF AUTOMATED TESTING IN ITER'S CONTROL SYSTEM SOFTWARE DISTRIBUTION*

R. Lange†, H. Kim, A. Zagar, ITER Organization, St. Paul lez Durance, France
M. Ruiz, V. Costa, J. Nieto, Grupo de Investigación en Instrumentación y Acústica Aplicada, Universidad Politécnica de Madrid, Madrid, Spain

Abstract

As part of the effort to standardize the control system environment of ITERs in-kind delivered >170 plant systems, the Controls Division publishes CODAC Core System (CCS), a complete Linux-based control system software distribution.

In the past, a large part of the integrated and end-to-end software testing for CCS was executed manually, using many long and complex test plan documents. As the project progress introduces increasing scope and higher quality requirements, that approach was not maintainable in the long term.

ITER CODAC and its partners have started a multi-year effort converting manual tests to automated tests, inside the so-called Framework for Integration Testing (FIT), which itself is being developed and gradually extended as part of the effort. This software framework is complemented by a dedicated hardware test stand setup, comprising specimens of the different controllers and I/O hardware supported by CCS. FIT and the test stand will allow to run fully scripted hardware-in-the-loop (HIL) tests and allow functional verification of specific software modules as well as different end-to-end use cases.

INTRODUCTION

The ITER project is a collaboration between seven members (China, Europe, India, Japan, Korea, Russia and the USA) representing 35 countries. Construction of the ITER Tokamak facility in southern France is largely (>90 %) based on in-kind procurement, i.e. the members developing and delivering hardware, components and systems to the project. That poses major challenges to the central ITER Organization, which is responsible for the specification, integration and operation of the machine.

Within the scope of the control systems aspect of CODAC (Controls, Data Access and Communication), the most important mitigation strategy is standardization. Hardware is standardized by limiting choices through strictly applying hardware catalogues. As part of standardizing software, ITER publishes and distributes a complete software distribution, based on Red Hat Enterprise Linux and named CODAC Core System (CCS) [1, 2].

Running this CCS software distribution is mandatory for all ITER control system-related computers.

Software Quality and Testing

Over the project's lifetime, the focus of CCS development activity has shifted. While the first versions were mainly adding features to reach the required functionality, recent versions were stabilizing the distribution, increasing its robustness and quality. With that shift, mandatory Quality Assurance (QA) tools have been introduced and software testing has gained importance.

Original concepts had foreseen the freezing of CCS software and plant system applications after site acceptance and integrated commissioning. As the project schedule evolved, it has become clear that continuing to update CCS is crucial to adapt to newer hardware and software, and to keep the Operating System parts under a support agreement.

THE EXISTING TEST ENVIRONMENT

On the front-end controller level, ITER maintains around 20 proprietary software modules, most of them Linux drivers and EPICS Device Support modules for I/O boards in the ITER hardware catalogue.

ITER software development follows a workflow based on ISO 12207. The testing required as part of this workflow is defined as a Software Test Plan (STP) document for each of the software modules. The procedures in these test plans are executed manually and results are entered into a Software Test Report (STR). Some STPs contain well over a hundred pages and take days to execute.

During the first years of development, most of the manual testing was done by external contractors, using a considerable amount of resources.

Changes Require a New Concept

With changes in the ITER project schedule and re-distribution of the budget, it became clear that this approach was not cost-effective and not sustainable.

To mitigate this, test coverage was cut down by only running a subset of tests and STPs with every release.

As a result of the limited testing, quality was decreasing. More bugs were found by users and had to be fixed and patched under time pressure.

AN AUTOMATED APPROACH

Automated Testing

Automated testing has always been used within CODAC Core System. The scope of automated testing, however, was limited to unit tests that are executed as part of the software module's build process.

* Work partially funded by PID2019-108377RB-C33/MCIN/AEI (Agencia Estatal de Investigación) /10.13039/501100011033 and PID2022-137680OB-C33/MCIN/AEI /10.13039/501100011033 / FEDER/ and the European Union.
† ralph.lange@iter.org

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM001

Extending the scope of automated testing needed a concept and an architecture. Starting in 2016, a concept for extending the test automation to cover complete modules end-to-end and integration testing of multiple modules was developed.

FIT – Framework for Integration Testing

The FIT concept describes a framework approach that:

- unifies testing approach, architecture and procedures,
- supports different testing levels and types,
- allows tests to be packaged and deployed on different test stands,
- supports the execution of hardware-in-the-loop tests by offering a hardware test stand,
- encourages developing tests that are run by scripts and can be automated,
- offers an automation component that can run tests as part of CI,
- can use test scripts that are easy to understand, change and extend,
- focuses on testing the APIs and the performance, not on the device HW accuracy,
- allows a test plan document to be extracted from the test script/documentation, and
- creates results in a format suitable for generation of proper test reports.

Tests are modular, being packaged in specific system packages (RPMs) that can be installed and run without having access to the sources and recompiling the software.

The software-under-test is always installed from their RPMs, never recompiled from sources, making sure that the tests are always reproducible and executed using the exact binaries that the users are installing.

Hardware-in-the-Loop (HIL) Test Stand

Dedicated 19”-racks contain a set of servers and specimens of the I/O hardware from the ITER catalogue, readily connected and available using a production-like network infrastructure. Inexpensive remotely accessible oscilloscope/waveform-generator/logic-analyzer instruments (using RedPitaya hardware [3] are added that allow generating analog and digital stimuli for inputs and recording signals from outputs.

Conceptually, the HIL test stand setup is kept as simple as possible:

- Connections between inputs, outputs and instruments are static. There is one setup that works for all tests.
- No connection switch box (matrix) is being used, as they have shown to require more cabling, a lot of maintenance and are hard to operate robustly.
- Tests are vertical, across all layers but separate for each module, avoiding cross-connections and cross-dependencies.

The goal is a HIL test stand that allows robust, reproducible operation of simple setups, allowing tests for different boards to be run in parallel.

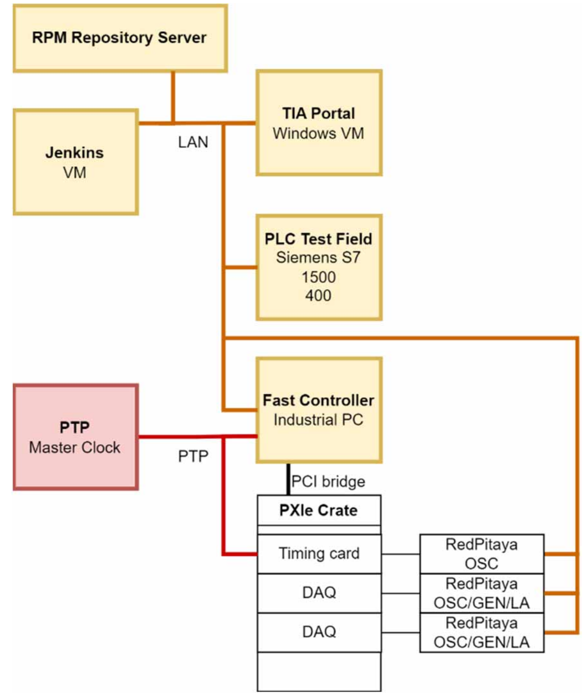

Figure 1: HIL Test Stand.

Figure 1 shows the logical architecture of the HIL test stand: Jenkins is used as the central automation engine. The test hosts are downloading all FIT tests and modules under test from the central RPM repository. Fast Controller(s) and PXIe timing cards are connected to the PTP master clock through a dedicated network. The RedPitaya instruments connect to the PXIe I/O cards. Regular LAN connects the instruments, the PLC test field and the TIA Portal VM that downloads the PLC test projects to the PLCs.

Figure 2: Test Stand Instruments Connect to I/O.

Figure 2 shows RedPitaya instruments connected to the I/O extension boards of the PXIe crate.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM001

FIT Test Module Standardization

The FIT test modules are regular CODAC Core System software modules, built and packaged using the normal workflow. When the resulting RPM is installed, it pulls the software under test (through RPM dependencies) and runs the tests.

Test scripts are written in Python, using a standard Python testing framework, like 'unittest', 'pytest' [4] or 'behave' [5].

The naming of tests and test procedures follows FIT conventions to allow CI jobs to run tests in a generic fashion.

CURRENT STATUS

In order to not exceed the budget limits, the HIL test stand had to be gradually built and equipped over the last five years.

Because of the high fluctuation in the ITER project staff and the limited manpower in the controls group, no progress on the software and framework parts was possible during that period.

In 2021, the Framework Contract for CCS Software Maintenance was awarded to a consortium including the Instrumentation and Applied Acoustic Research Group at the Technical University of Madrid – the same team that developed the scripted testing of the different NDSv3 software modules, reported at the previous ICALEPCS conference [6]. This team with their excellent knowledge of the CCS system and their experiences with scripted HIL testing is the perfect partner to start working on the implementation of the FIT software parts.

A five-year project has been started in 2021, covering the migration of the mentioned manual test plans towards FIT test modules. The framework parts of FIT are being developed along with the test modules, mainly by moving parts and functions that are shared between test modules into the framework. Similarly, the HIL test stand gets extended and improved as more tests are moved towards FIT.

The order of the tests that are being converted is driven by their complexity (from simple to complex tests) and the expected saved effort (from tedious, large but "stupid" tests to more sophisticated procedures).

Not all existing manual tests procedures can be scripted. In some cases, the approach is to step back to the requirements that need to be verified and create a new set of scripted test procedures. In other cases, remaining manual tests are moved into a "manual" test plan that gets executed in addition to the scripted tests.

At this point, the following FIT modules are implemented:

- **fit-instruments** for supporting the interface with the instruments used for signal generation and measurement. This module includes the support for RedPitaya and oscilloscopes from Tektronix and Keysight. SCPI commands are used to standardize the communication with the instrument.
- **fit-nisync** for the automatic testing of the API of the National Instruments (NI) PXI6683H board, which manages timestamps and future time event generation of different digital lines (external or on the PXI Trigger bus).
- **fit-xseries** for automatically testing the API of the NI X-Series devices PXIe6363 and PXIe6368. The test validates the different hardware elements, including the advanced trigger functionalities.
- **fit-s7plc** for the automatic testing of the EPICS S7 PLC Device Support that ITER has implemented. The FIT module uses a Python client for the OPC UA protocol to communicate directly with the PLC and validate the operations executed by the EPICS Device Support.
  An additional custom Python module has been implemented to communicate with a Windows computer to download specific PLC test projects to Siemens PLCs using the TIA Portal connection.
- **fit-ptpd-tcn** for automatically testing the ITER ptp daemon in different hardware interfaces (PXI6683H and i350 Ethernet interfaces) and the TCN (timing network) library.

The implementation of these FIT modules reuses and adapts some existing software applications of the modules under test that were developed for manual testing, such as tests based on the Google test library and simple example programs. Other manual tests needed to be rewritten to remove any operator intervention. In all cases, the software tools used are free and open-source.

Removing the operator intervention guarantees the repeatability of the tests and allows to reproduce the results on different hardware platforms easily.

The time needed to execute the tests has been dramatically reduced. (Specifically, the test of the EPICS Device Support for Siemens S7 PLC has been reduced from two weeks to less than two hours.) Repeating the test execution hundreds or even thousands of times improves finding hidden problems in the device drivers or the hardware firmware.

NEXT STEPS AND FUTURE PLANS

Over the coming years, more test plans will be migrated, including test fixtures spanning multiple machines that need coordination and orchestration between the parts running on different hosts.

Linux container technology (Docker/Podman) will be introduced to allow testing on different CCS releases without rebooting or reinstalling the test host(s).

Jobs will be added to CODAC's internal CI service, based on Jenkins, to allow running tests periodically or triggered by software commits and the following rebuild of RPM packages.

Running the tests many times will need specific software tools to analyze the obtained results and create summary reports to determine the stability of the software APIs and the hardware elements under test.

An obsolescence management plan is needed to guarantee the evolution of the FIT system regarding to changes in hardware devices under test, the software elements of CCS,

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM001

and the hardware and software elements used in the test instrumentation (RedPitaya, oscilloscopes, etc.).

CONCLUSION

In a situation where the established manual testing procedures for CODAC Core System software could not be sustained any longer, a new concept based on automated testing was put in place and a multi-year project of migrating tests to the new concept was launched.

After two years, the first set of converted tests show the validity of the concept and give an impression of the possible testing time and resource savings.

REFERENCES

[1] F. Di Maio *et al.*, "The CODAC Software Distribution for the ITER Plant Systems", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper MOPKS029, pp. 227-230.

[2] A. Wallander *et al.*, "News from ITER Controls – a Status Report", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper MOBAUST01, pp. 1-4.

[3] Red Pitaya – Swiss Army Knife for Engineers, `https://redpitaya.com/`

[4] pytest: helps you write better programs, `https://docs.pytest.org/`

[5] behave: behavior-driven development, Python style `https://behave.readthedocs.io/`

[6] R. Lange *et al.*, "Nominal Device Support (NDSv3) as a Software Framework for Measurement Systems in Diagnostics", in *Proc. ICALEPCS'21*, Shanghai, China, Oct. 2021, pp. 337-342. `doi:10.18429/JACoW-ICALEPCS2021-TUBR01`