AUTOMATED TESTING OF ITER DIAGNOSTICS SCIENTIFIC INSTRUMENTATION

Stefan Simrock
ITER Organization
St. Paul-lez-Durance
stefan.simrock@iter.org

Arthur Leveque (Arthur.Leveque@iter.org)[1], Lana Abadie (Lana.Abadie@iter.org)[1], Bertrand Bauvir (Bertrand.Bauvir@iter.org)[1], Robin Barnsley (Robin.Barnsley@iter.org)[1], Luciano Bertalot (Luciano.Bertalot@iter.org)[1], Petri Makijarvi (Petri.Makijarvi@iter.org)[1], Ralph Lange (Ralph.Lange@iter.org)[1], Mikyung Park (Mikyung.Park@iter.org)[1], Roger Reichle (Roger.Reichle@iter.org)[1], Denis Stepanov (Denis.Stepanov@iter.org)[1], George Vayakis (George.Vayakis@iter.org)[1], Anders Wallander (Anders.Wallander@iter.org)[1], Pierre Petitpas (Pierre.Petitpas@iter.org)[1], Michael Walsh (Michael.Walsh@iter.org)[1], Zhao Li (zhaoli@swip.ac.cn)[2], Tsuyoshi Yamamoto (yamamoto.tsuyoshi@qst.go.jp)[3], Sanjeev Varshney (sanjeev@iter-india.org)[4], Jihyun Choi (jhchoi80@nfri.re.kr)[5], Ekaterina Mironova (e.mironova@iterrf.ru)[6], Paul Sichta (psichta@pppl.gov)[8], Prabhakant Patil (Prabhakant.Patil@iter.org)[9], Manojkumar Annigeri (Manojkumar.annigeri@tcs.com)[9], Sukhmeet Singh Kalsi (Sukhmeet.Kalsi@iter.org)[9], Dariusz Makowski (dmakow@dmcs.pl)[10], Piotr Perek (pperek@dmcs.pl)[10], Mariusz Orlikowski (mariuszo@dmcs.pl)[10], Kevin Meyer (kevin.meyer@cosylab.com)[11], Vincent Martin (Vincent.Martin@iter.org)[12]

[1]ITER Organization, St. Paul-lez-Durance, [2]ITER China Office, Beijing, China, [3]Japan Atomic Energy Agency, Ibaraki, Japan, [4]ITER-India, IPR, Gandhinagar, India, [5]ITER-Korea, Yuseong-Gu, Daejeon, Republic of Korea, [6]ITER-Russia, Russian Research Center "Kurchatov Institute", Moscow, Russian Federation, [7]Fusion for Energy, Barcelona, Spain, [8]PPPL, Princeton, United States, [9]Tata Consultancy Services, Pune, India, [10]DMCS, Lodz University of Technology, Poland, [11]COSYLAB, Lubljiana, Slovenia, [12]BERTIN Technologies, Aix-en-Provence, France

Abstract

ITER requires extensive diagnostics to meet the demands for machine protection, machine operation, plasma control, evaluation and physics studies. Many of the diagnostics require high performance scientific computing for the execution of complex algorithms for the measurements. Each diagnostic system has a dedicated plant I&C system, which must meet stringent requirements of latency, data throughput, reliability and availability. Hence, the testing and acceptance of diagnostic plant I&C systems is complex. Also due to large number of diagnostic systems which require plant I&C, it is important that the design, implementation and testing follow a common methodology. The paper describes the diagnostic plant I&C development methodology followed at ITER in brief. The overall strategy and methodology for testing and acceptance of diagnostic plant I&C system is discussed in detail. The approach and implementation of automated testing of diagnostic plant I&C system is also presented.

1. INTRODUCTION

This ITER ("The Way" in Latin) is one of the most ambitious energy projects in the world today. The aim of this project is to build the world's largest tokamak, a magnetic fusion device that has been designed to prove the feasibility of fusion as a large-scale and carbon-free source of energy based on the same principle that powers our Sun and stars. ITER requires diagnostic systems to perform measurements for machine protection, machine operation, plasma control, evaluation and physics studies. The different families of diagnostic systems to be developed are magnetic, neutron, optical, bolometric, spectroscopic and microwave. In addition, plasma facing and operational diagnostic systems aid in the protection and operation of the machine.

Many of the more than 50 diagnostic systems require high performance scientific computing for the execution of complex algorithms related to measurements. The most stringent requirements include high performance data acquisition, data processing, real-time data streaming from distributed sources to the plasma control system and large amounts of raw data streaming to scientific archiving system. While most of these requirements have been achieved individually, the challenge for ITER will be the integration of these state-of-the-art technologies in a coherent design while maintaining all of the performance aspects simultaneously. The testing and calibration of the diagnostic systems in the laboratory or on existing machines is also vital.

The ITER Instrumentation and Control (I&C) system enables integrated and automated operation of ITER. It has two levels of hierarchy i.e. central I&C systems and plant I&C systems. The central I&C systems coordinate and orchestrate all plant I&C systems, including plant-wide investment protection and safety functions and to provide the human-machine interface (HMI). A plant I&C system controls a plant system including local investment protection and safety functions. In order to facilitate integration, plant I&C systems have been grouped into control groups. The control groups are in turn connected to the central I&C systems.

Each diagnostic system has a dedicated plant I&C system and these plant I&C systems are grouped under diagnostic control group. The diagnostic plant I&C system must meet around 500–700 functional and non-functional requirements which include also the requirements from the ITER handbooks such as the Plant Control Design Handbook (PCDH), Electrical Design Handbook (EDH) and the Nuclear Radiation

Compatibility Handbook. For the development of diagnostic plant I&C systems a methodology based on Systems Engineering has been developed.

2. DIAGNOSTICS PLANT I&C DEVELOPMENT METHODOLOGY

The Systems Engineering for diagnostic plant I&C is applied in accordance with the principles established by the International Council On Systems Engineering (INCOSE) [1],[2],[3] with the underlying industrial standard described in ISO/IEC/IEEE 15288 [4]. Enterprise Architect (EA) is used as the system design engineering tool. Dedicated EA add-ins supporting the methodology have been developed so that design documentation can be automatically generated. CODAC Core System (CCS) software package, which is distributed by ITER is used for the development of diagnostic plant I&C system. The diagnostic plant I&C development methodology includes the following design elements:

Use Cases and Requirements – The requirements for diagnostic plant I&C are captured as use cases for the whole diagnostics, requirements from the ITER handbooks (PCDH, EDH, Operations Handbook, Radiation Compatibility) and design inputs specific to each diagnostic from sub-System Requirement Document (sub-SRD). The typical use cases include measurements, calibration, maintenance, acceptance testing, commissioning and integration, signal chain integrity checks and configuration management. The requirements are categorised as functional, non-functional, safety and interlock requirements. The requirements are further broken down into multiple levels as per the requirement breakdown structure shown in Figure 1.

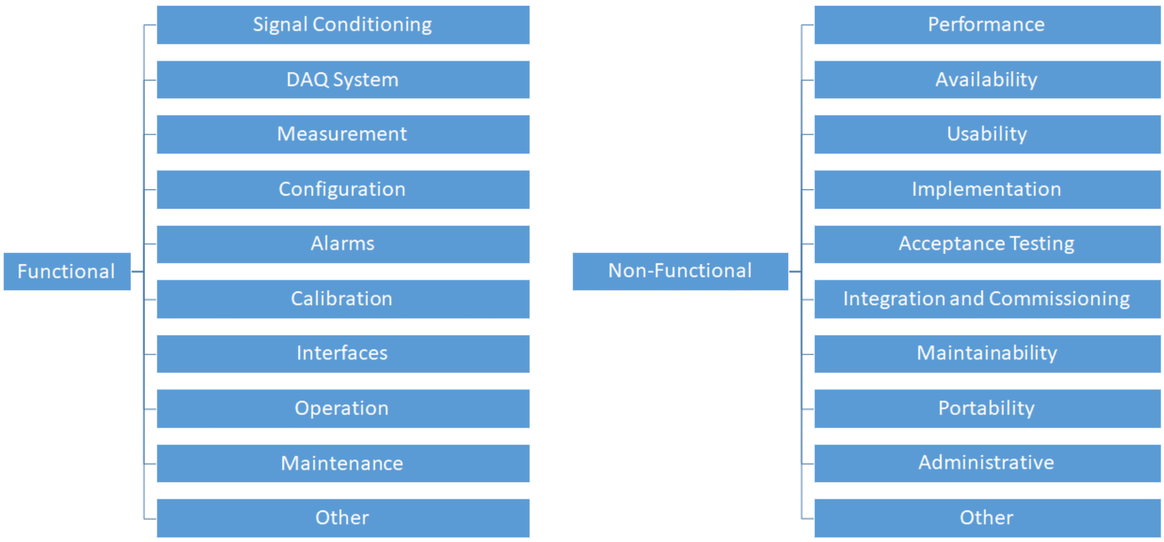

Figure 1. Requirement Breakdown Structure Level 2

Procedures – The procedures for tokamak operation, expert operation, Factory Acceptance Testing (FAT), Site Acceptance Testing (SAT), integrated commissioning, maintenance, system conditioning, plant protection, occupational safety and exception handling are defined using activity diagrams. The procedures are broken down into multiple levels as per the procedure breakdown structure.

Functional analysis – Functions are required for the execution of procedures. Functional analysis is the process of identifying and describing the functions. Functions cannot be derived by drilling down the procedures per se, instead many views used in all lifecycle phases must be used during functional analysis. Functional breakdown into multiple levels helps in definition of functions. Typical functions at level 1 include signal conditioning, data acquisition, data processing, diagnostics measurements, system management, operation, machine protection, occupational safety, Commercial off-the-shelf (COTS), interface and global. At the lowest level of functional breakdown are the variables, defining function inputs/outputs.

Hardware architecture – The hardware architecture defines the CPU/network chassis, I/O chassis, High Performance Network (HPN) interface boards and I/O cards required to implement all the functions described in the functional analysis. Signals associated with sensors and actuators are defined and variables are mapped to signals. Bill of materials and cubicle internal wiring diagram is also prepared.

Automation – State machines are defined to describe the automation for a given diagnostic plant I&C. This includes the conditions for state transitions and entry-do-exit procedures. To synchronise all diagnostic plant I&C systems during operation through central I&C system Common Operating States (COS) and Plant System Operating States (PSOS) are used. The mapping of PSOS states to COS states is done for each diagnostic plant I&C system.

Important aspects of design maturity assessment are the functions to requirement mapping and procedures to function mapping. A typical development workflow for diagnostic plant I&C system is shown in Figure 2. The various tools and repositories used are also depicted.

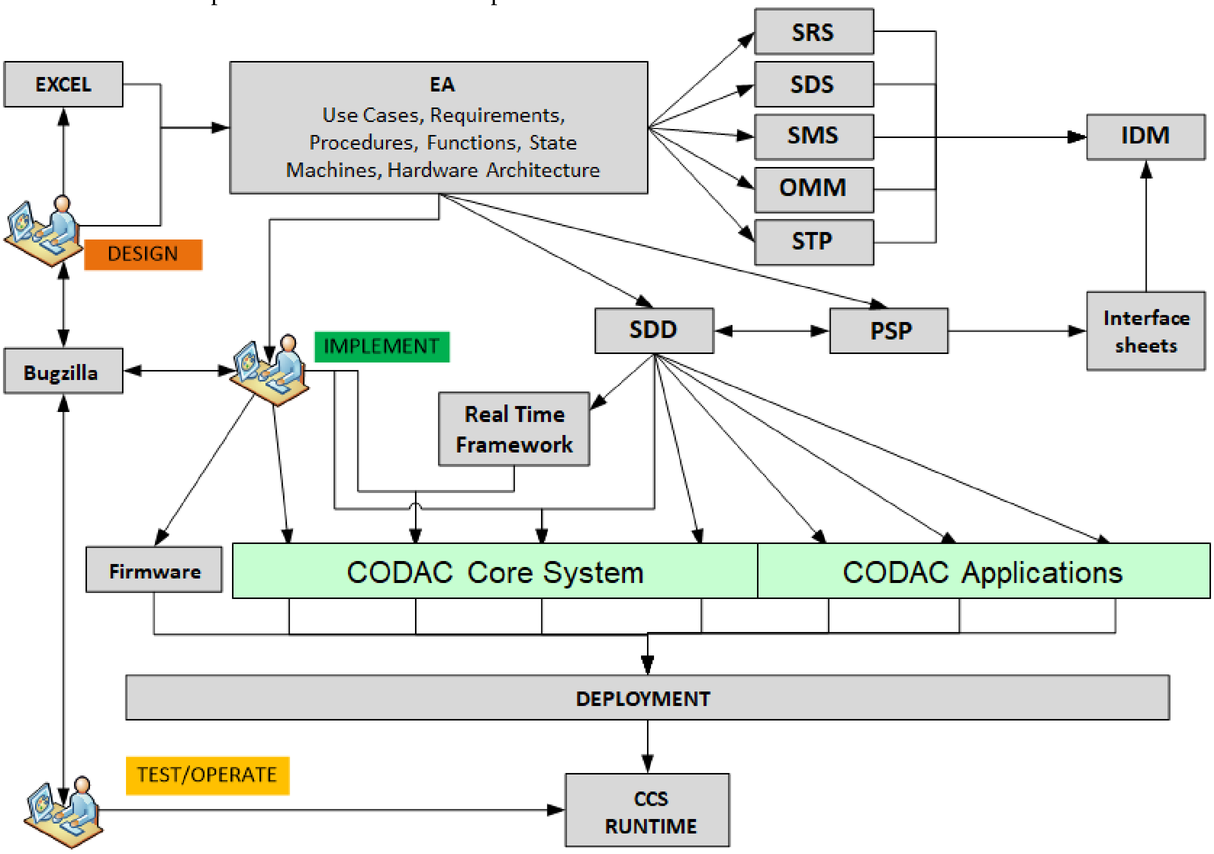

Figure 2. Workflow for Plant I&C Development

Use cases, requirements, procedures, functions, variables, signals, components, hardware architecture and state machines are defined in EA. Deliverables for diagnostic plant I&C design i.e. System Requirement Specification (SRS) and System Design Specification (SDS) are generated automatically from EA and stored in ITER's Document Management System (IDM). EA has interface with CCS's SDD toolkit and Plant System Profile (PSP) database. Interface sheets can be generated directly from PSP.

Real Time Framework (RTF) provides the infrastructure to facilitate implementation of real-time tasks within diagnostic plant I&C system and derive diagnostic measurements. The Plasma Control System (central I&C system) is also implemented in RTF. CCS software package along with RTF forms the core for development of diagnostic plant I&C systems. Bugzilla has been setup to keep track of reported bugs and known issues.

3. TESTING AND ACCEPTANCE OF DIAGNOSTICS PLANT I&C SYSTEMS

The diagnostic systems require large and dynamic measurement range; low latency and high data throughput. Measurements from multiple diagnostic systems are often combined to derive diagnostic parameters. Diagnostic systems require complex calibration procedures. For example, for the Plasma Position Reflectometry diagnostic system the calibration requirements include calibration of internal delay line over whole frequency range. This requires dedicated test equipment which is part of diagnostic plant I&C system. Reliability and availability requirements for ITER diagnostic systems are also very stringent. For example diagnostic sets providing measurement parameters for Machine Protection shall be designed for a combined online failure rate of less than $5*10^{-5}$ per measurement per operational (calendar) year.

Because of the aforementioned reasons, in contrast to industrial systems where the tests usually evaluate a sequence of simple device functions, the testing and acceptance of diagnostic systems is much more complex. The validation of integrity of signal/ data processing chain in diagnostic plant I&C system is of utmost importance.

3.1. Overall Strategy

The acceptance of diagnostics plant I&C system is based on the compliance of the system with the requirements. The compliance is evaluated through execution of a test plan which is based on various test scenarios in which a series of tests are executed. The result of the tests are determined by specific pass/fail criteria and are documented in System Test Report (STR). A summary of the test results are given in a compliance matrix.

Testing of diagnostics plant I&C will be conducted at different life-cycle stages of the project namely

— FAT : Factory acceptance testing of diagnostic plant I&C system components
— SAT : Site acceptance testing of a complete diagnostic plant I&C system
— Integrated commissioning : Testing of the diagnostic plant I&C system integrated with central I&C system
— Operation: On-line testing of system components in context of operation
— Maintenance : Troubleshooting (detecting defects), repair (replacement of hardware, up-date/upgrade of software), re-testing for validation before system is released for operation

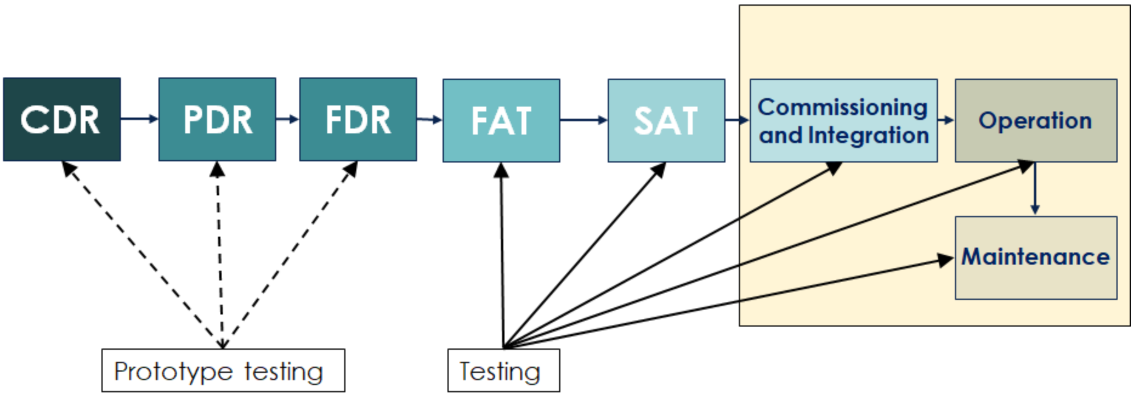

Figure 3. Testing in Various Life-cycle Phases

In each phase, functional tests will be carried out to ensure that all relevant functions (as specified in the SRS and SDS) can be executed successfully. These are the core tests performed for validating the compliance with design requirements. Procedure tests validate that all required functions needed for operation are available.

The main types of test that will be conducted are:

— <u>Unit and components tests</u> are low-level tests that ensure that the essential components in a system meet the requirements as a pre-requisite for system tests. While unit tests require a dedicated test environment (usually performed by the manufacturer of a component), the component tests are performed in the context of the system in which they will be used (normal operation). In other words, component tests are similar to unit tests except that real data instead of dummy data is used for testing.
Testing a data acquisition board by the manufacturer is an example of unit test. Examples of component tests include tests for Data Archiving Network (DAN) and Synchronous Databus Network (SDN) for a diagnostic plant I&C system.
— <u>Integrated tests</u> verify that the different components are working well together in diagnostic plant I&C system. The objective is to demonstrate that the diagnostic plant I&C system is functioning as an integrated system.
— <u>Integration tests</u> verify that the diagnostic plant I&C system is correctly interfaced with the central I&C systems and that the diagnostic plant I&C system can be operated from the central I&C system. These tests are a pre-requisite before performing the integrated commissioning, in which all procedures for operation of diagnostic plant I&C system are executed.
— <u>End-to end tests</u> demonstrate the usability of the diagnostic plant I&C system for all applications the system has been designed for. This means that all procedures derived from the system use cases can be executed using the plant I&C functions to the maximum extent possible (i.e. only a minimum of non-I&C or external functions are required). During these tests, the interaction with other plant I&C systems is also evaluated.

Testing must be supported by built-in diagnostic plant I&C functions that can be executed from the central I&C system using central applications and services. For example to perform performance tests, diagnostic plant I&C must support functions which can measure the performance value. Typical performance measurements are related to data throughput and latency, data error rates, system availability, data processing loads, and timing accuracy.

Some steps in the test procedures may have to be executed using non-I&C functions (such as switching on power to a chassis or observing a light indicator on I&C equipment) or using I&C functions of the central

CODAC system (such as a linux commands from a terminal to compile source code or start an IOC) but the number of these functions should be minimized.

3.2. Test Plan

System Test Plan (STP) (separate for each life-cycle phase) for diagnostics plant I&C explains the various test scenarios which need to be executed. The various test scenarios for SAT are (i) environment (visual inspection, cable connectivity, hardware connectivity), (ii) installation, (iii) network configuration and connectivity, (iv) hardware configuration and connectivity, (v) unit/component, (vi) integrated system, (vii) performance, (viii) operation, (ix) automation and (x) load and stress. The pass fail criteria, suspension criteria and resumption requirements are also defined in the STP.

3.3. Test Execution

Test execution is the process of executing the tests and comparing the expected and actual results. The following steps are typical for the execution of a test:

1. Selection of the test scenario, specific test cases and related test procedures.
2. Configuration of the system under test according to the test procedures.
3. Verification that the pre-requisites for the test are fulfilled (check lists).
4. Execution of the test procedures.
5. Resolve blocking issues as they arise or terminate the test if indicated.
   Note: In case of test termination maintenance must be scheduled.
6. Report test findings and status. Document test results and manage problems in issue tracker.

3.4. Test Report

The test results are documented in STR. The summary of the test results is depicted through a compliance matrix as shown in Figure 4. It represents the coverage of functions and requirements by the test results.

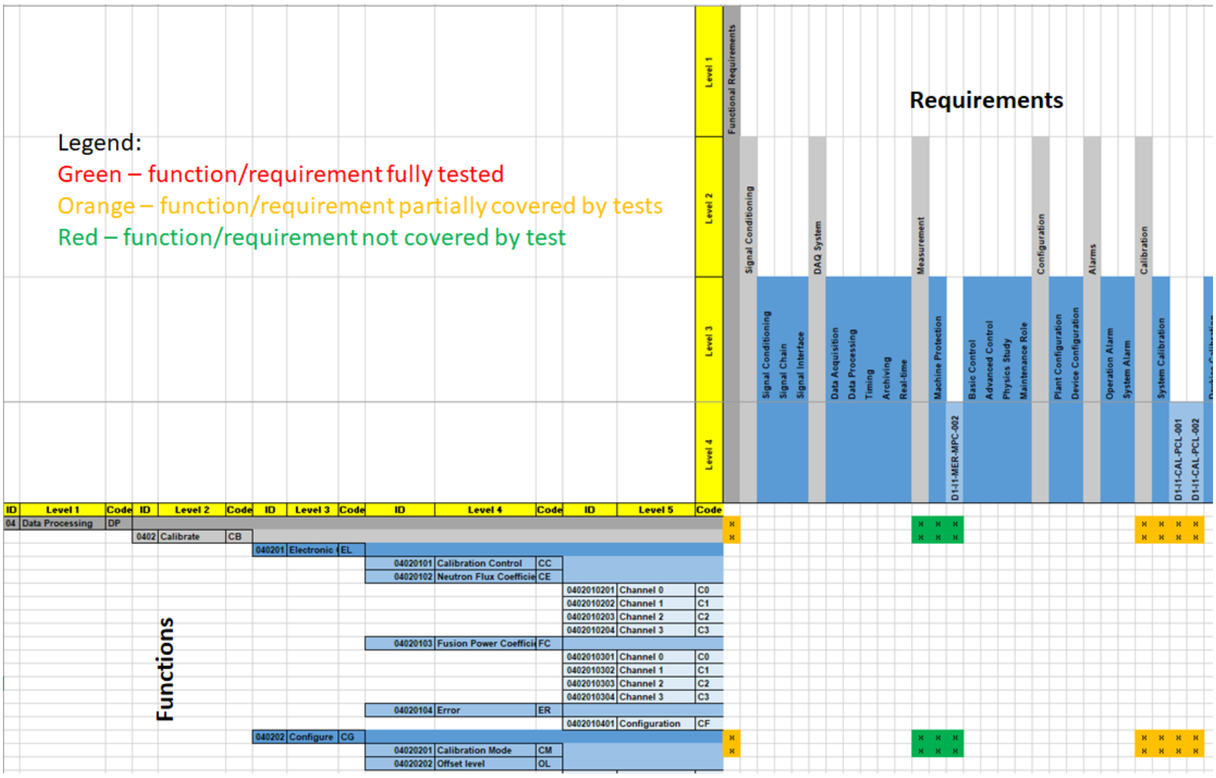

Figure 4. System Compliance Matrix

4. APPROACH FOR AUTOMATION TESTING

Manual testing is performed by a human sitting in front of a computer carefully executing the test procedures. In contrast, automation of testing replaces the human using automation tools (like scripts) to execute the test procedure. As a minimum, test automation assists the humans through the use of tools, scripts and specialized test software.

Automated testing must be distinguished from test automation, which refers to automating the process of tracking and managing the different tests. It makes use of special software (separate from the software being tested) to control the execution of tests and the comparison of the outcomes with predicted outcome.

4.1. Motivation for automated testing

Automated testing is important for the following reasons

- Manual testing is very time consuming.
- It is difficult to manage manually the complex configuration for testing.
- Manual testing can become boring and hence error-prone.
- Automated testing does not require human intervention. One can run automated test unattended (overnight). This increases the reliability, consistency and accuracy of test results.
- Automated testing increases the speed of test execution thus saving time and cost.
- Automated testing helps increase test coverage
- Automated testing increase system availability

4.2. Scope of Automated Testing

The scope of automated testing is the automation of function and procedure testing.

Function testing - Functions are evaluated through their input-output relationship and pass/fail criteria are applied to determine the compliance with the requirements. Function tests can be implemented in the ITER Control System Framework since the functions are accessible through their variables. Functions can be tested at the lowest level where the functions have only a few input/output variables and the input-output relationship is described by a simple algorithms. Usually, a group of functions which are closely related to each other are tested together.

Since the functions are tested in context of the diagnostic plant I&C, it is necessary to configure diagnostic plant I&C according to the selected test case. The inputs of some functions are produced by live measurements and can therefore not be directly controlled by the control system. In this case, it is necessary to ensure that the operation conditions of the plant system are adequate for the test to be performed. An example of a function to be tested in the signal processing chain is shown in Figure 5. For the pulse discrimination function, the input variables are configuration parameters and raw data and the output variables are the measurements and meta-data. The pulse discrimination function passes the test if the measured input – output relationship matches the design expectations for a pre-determined range of input parameters.

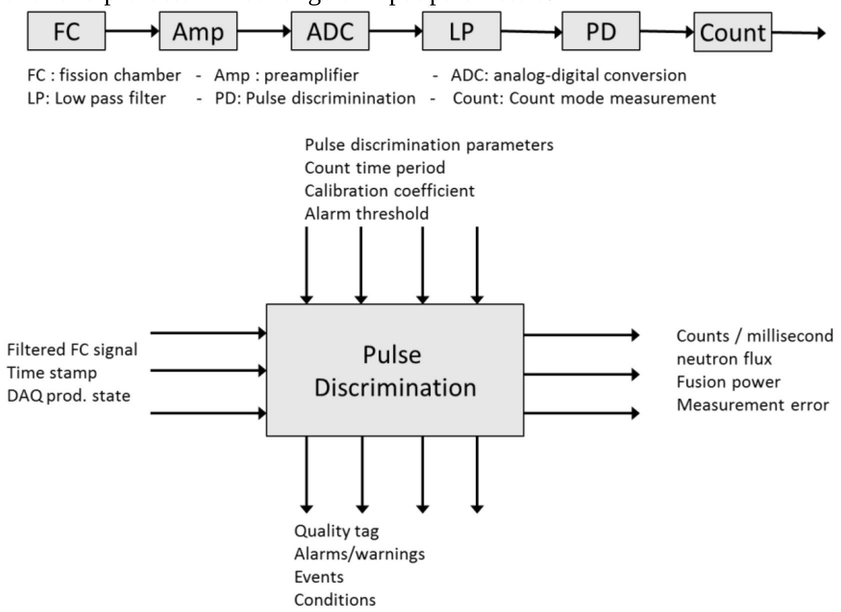

Figure 5. Pulse Discrimination Function in Neutron Flux Measurement Diagnostics

Procedure testing - At the lowest level procedures are decomposed in a few steps which in the ideal case can be executed through a sequence of diagnostic plant I&C functions. It is important to note that a procedure may still pass in presence of a diagnostic plant I&C function failure.

A procedure is initially evaluated through the execution of its main flow which represents the normal operation. For this, it is important that all necessary conditions for the procedure are fulfilled. The set-up of complex conditions may require the execution of preparatory procedures to establish these conditions. Once the main flow of a procedure has been successfully executed, the procedure is said to have passed. After testing the main flow, the branching flows of the procedure are tested. For this one has to introduce the necessary error conditions required for the branching.

4.3. Framework for automated testing

The framework for automated testing will be the same as for automation of diagnostics plant I&C operation. Automation can be implemented using state machines or scripts. State machines are usually used if (complex) dependencies have to be managed during testing. State machines will be implemented using State-Notation-Language (SNL).

EPICS on which the Codac Core System (CCS) is based, provides various choices of programming languages for developing channel access clients. These include Perl, Matlab, Python, Java and C#. Python offers several convenient features for scientific and engineering programming such as clean syntax, cross platform, high level, object oriented, easily extensible and availability of many libraries. In addition, Python contains many scientific libraries like numpy and scipy. Hence, Python (using channel access (CA) and process variable access (PVaccess)) is chosen as the language for developing test automation scripts. Central tools/applications and repositories like configuration tool, central supervision, pulse schedule editor and pulse schedule validation will also be used.

5. IMPLEMENTATION OF AUTOMATION TESTING

The functional tests of the reference diagnostic plant I&C system developed at ITER have been executed using python unit test framework unittest2. The functional tests related to data acquisition include tests for initialization for data acquisition, raw data acquisition for different channels, clock divisor of ADC device, ADC device sampling rate, data source selection, data source configuration error, hardware triggers, software triggers and filter frequency. Figure 6 shows the Python code snippet for data acquisition functions test. Figure 7 shows the execution of data acquisition functions test.

```python
def test_01_RawData_Channel03(self):
        """/RAW Data/Channel [0-3]
           Raw data channel [0-3] acquisition verification with test sine wave 1.25MHz (Amplitude 0.998V expected)
           Tested functions: DQADRDC000 DQADRDC100 DQADRDC200 DQADRDC300 """
        for c in [0,1,2,3]:
            variable = epics.PV('D1-I9-BCA0:DQADRDC%d00-MROWR' % c, auto_monitor=False)
            val = variable.get()
            self.assertEqual(variable.status, 0, msg='[channel %d]' % c)
            self.assertEqual(variable.severity, 0, msg='[channel %d]' % c)
            self.assertAlmostEqual(max(val)-min(val), 0.998, msg='[channel %d]' % c, delta=0.05)
```

Figure 6. Code Snippet for Data Acquisition Functions Test

```
====================
Test Case:  B_DataAcquisition_ADConversion.test_01_RawData_Channel03
Description:
            Data Acquisition/ADC Conversion/RAW Data/Channel [0-3]
            Raw data channel [0-3] acquisition verification with test sine
wave 1.25MHz (Amplitude 0.998V expected)
Tested functions: DQADRDC000 DQADRDC100 DQADRDC200 DQADRDC300
Testing ...
PASSED
```

Figure 7. Data Acquisition Functions Test Execution

Similarly, functional tests are performed for data processing, system management, interfaces and algorithms using scripts. The procedures which can be automated and required no operator intervention are also tested using scripts. Example of such procedure is Produce and Archive DAN, SDN and PON data. For the procedures requiring operator intervention scripts based on state machine are used, which check for a large set of conditions.

Self-Description Data (SDD) is an ITER concept designating the static data that configures the plant system I&C. This data is produced by the plant system I&C designers and developers, using IO tools and according to an IO defined schema. The CCS software package distributed by IO CODAC section for the development of Plant I&C contains Self-Description Data toolkit. The toolkit contains SDD editor, SDD translator, SDD sync and SDD parser. The SDD tools are vital for configuring plant systems as per the test campaign being run.

State management is realized in CODAC using common operational state (COS) and plant system operational states (PSOS). COS-PSOS mapping is used to synchronize all plant systems for the execution of a pulse. EPICS PVs are used for communicating requests and states between CODAC and plant systems. State machines are implemented using State Notation Language (SNL) in plant systems. For testing these state machines COS-PSOS OPI is used, which is automatically generated by SDD editor from the COS-PSOS mapping. Figure 8 depicts the COS-PSOS OPI.

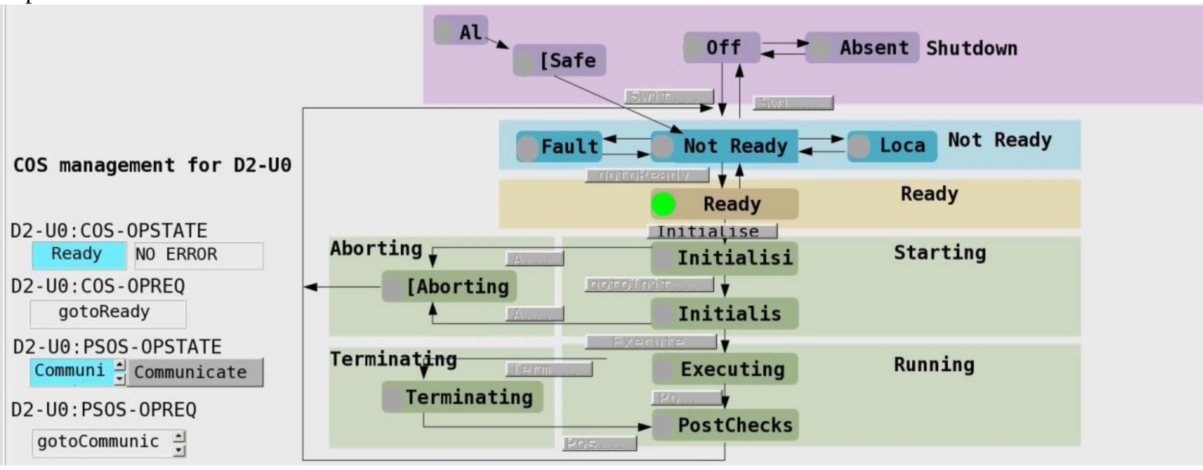

Figure 8. COS-PSOS OPI

For state change, tester presses the corresponding button in the COS-PSOS OPI. In Figure 8, tester can press gotoReady button to go to Ready state. If all the conditions for going to Ready State are met, the plant system goes to Ready state.

6. CONCLUSION

Testing of diagnostic plant I&C is a complex process required in all life-cycle phase and needs to be repeated regularly during operation and following every maintenance activity. The manual execution of the tests required for a complete system verification is very time consuming and can take several days. Therefore automated testing is essential to reduce the time and cost of testing to a minimum. Additional benefits of automated testing are : (1) wider test coverage, (2) reliable results, (3) improved consistency and accuracy, (4) human intervention is minimized, (5) increased efficiency at higher speed, (6) re-usable scripts, and (7) more frequent testing. This results in earlier deployment of the diagnostics plant I&C and improved reliability and availability.

ACKNOWLEDGEMENTS

Special acknowledgement is given to all IO and DA team members involved in the design of the ITER diagnostic systems and CODAC.

DISCLAIMER

The views and opinions expressed herein do not necessarily reflect those of the ITER Organization.

REFERENCES

[1] INCOSE, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th ed., San Diego, CA, USA, Wiley, 2015

[2] INCOSE UK, Z-Guides

[3] BKCASE Editorial Board. 2017. The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 1.8. R.D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology

[4] ISO/IEC/IEEE 15288:2015, Systems and software engineering - System life cycle processes.