ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA098

DEVELOPMENT OF A PXI BASED TEST STAND FOR AUTOMATIZATION OF THE QUALITY ASSURANCE OF THE PATIENT SAFETY SYSTEM IN A PROTON THERAPY CENTRE

P. Fernandez Carmona, M. Eichin, R. Van der Meer, F. Heimann, H. Regele, M. Grossmann, D. Weber, Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

Abstract

At the Centre for Proton Therapy at the Paul Scherrer Institute, a cyclotron, two gantries and a fixed beamline are being used to treat tumours. In order to prevent non-optimal beam delivery, an interlock patient safety system (PaSS) was implemented that interrupts the treatment if any sub-system reports an error. To ensure correct treatment, the PaSS needs to be thoroughly tested as part of the regular quality assurance as well as after each change. This typically required weeks of work, extensive beam usage and could not always cover all possible failure modes. With the opportunity of the installation of a new gantry, an automated PaSS test stand was developed that can emulate the rest of the facility. It consists of a NI PXI chassis with virtually unlimited IOs that are synchronously stimulated or sampled at 1MHz, a set of adapters to connect each type of interfaced signal and a runtime environment. We have also developed a VHDL based formal language to describe stimuli, assertions and specific measurements. We present the use of our test stand in the verification and validation of the PaSS, showing how its full quality assurance, including report generation was reduced to minutes.

INTRODUCTION

At the Paul Scherrer Institut (PSI) cancer patients are being treated using proton therapy for a number of indications. The facility currently includes a fixed beam line for eye cancer treatment, operating clinically since 2010, and Gantries 1 and 2 operating since 1996 and 2013 respectively [1, 2]. A dedicated 250 MeV cyclotron is used to provide beam for all the treatment areas. Recently a new Gantry 3 has been installed and is being commissioned. Both the latter Gantry and the accelerator are commercial products from the company Varian Medical Systems [3]. The rest of the treatment areas were designed in house.

Each for the treatment areas designed at PSI, as well as the adapter used for the integration of Gantry 3 include a Patient Safety System. PaSS is the system responsible to monitor the status of the different elements involved in the treatment and to stop the beam to avoid personal harm whenever any potentially unsafe condition is detected.

The Patient Safety System needs to be thoroughly tested in order to ensure correct treatment. The quality assurance first includes testing the monitors and final elements connected to it. This is typically part of the commissioning process and regularly scheduled QA tests [4]. Secondly the hardware undergoes unit testing. This involves a preparation phase, when a risk analysis is performed and a test specification based on the design specifications are written, and an execution phase. The execution is both performed as a simulation, and later physically tested in the lab with a test stand that stimulates all inputs and monitors all outputs and checks that the response is as expected. Thirdly an integration test in the facility is performed, when all supervision functions and all final elements are checked for correctness, and errors are injected to monitor the PaSS response. It is important to note that being executed in a clinical facility, not all cases can be covered in this last step.

UNIT TESTING CONCEPT

The unit testing consists of a series of test steps applied to the PaSS system to emulate real life conditions at the interface level. Each of the unit tests are derived from different aspects of the design specifications and specify both a stimulus to be applied to the input signals and an expected behaviour of the output signals. They are described in a document that describes the test using and timed signal diagrams.

With the introduction of Gantry 3, a new unit testing methodology was introduced and it is now also being gradually applied to the other treatment areas. The unit tests are specified in a formal language that was developed for this task and which will be detailed in the following section. This reduces the amount of manual work, removes the ambiguities of natural language and therefore minimizes the possibility of errors. Also, thanks to the technical progress of instrumentation hardware with a high count of fast digital IOs, such as National Instruments´ PXI crates, it is now possible to synchronously all input signals and sample all output signals instead of sequentially testing small subsets of signals, as was the case in our former unit testing setup.

The three main aspects described in the unit tests are the stimuli, expected reaction and time measurements. Stimuli can be both realistic as well as physically impossible in the real facility. The expected and forbidden reactions are described as logic assertions. A number of time measurements can be programmed to evaluate the performance of the hardware and its logic.

UNIT TEST FORMAL DESCRIPTION

After an investigation of different existing languages to describe tests and assertions, nothing was found that was both compact and close enough to natural language as to be able to replace the textual description in the unit test

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA098

documentation. The best option for our needs was VHDL syntax, which was extended to provide two missing aspects: Macros and expansion loops.

An example code of sequential stimuli would be:

```
SENSOR_A     <= NOK;
wait for     t_Response;
ELEMENT_B    <= OK;
```

Assertions are conditions that need to be true, otherwise an error of a certain severity is thrown:

```
assert STATUS_C = NOK report " STATUS_C not as expected" severity FAILURE;
```

Time measurements can be specified to evaluate the performance of certain elements:

```
measure falling_edge(SIGNAL_1) to rising_edge(SIGNAL_2) name “Example measurement";
```

The unit tests can be implemented in a single or multiple files. Macros can be defined in files containing typical time constants or reusable functions that can be included and called from several unit tests.

```
DefineMacro TYPICAL_TIME_CONSTANTS
constant t_Response  : time := 50 us;
EndMacro
```

Expansion loops are a custom extension used to repeat a same unit test under different conditions or variations. A typical effort saving case would be to check a certain function under experimental and therapy modes, in combination with the allocation or non-allocation of mastership.

```
Process Stimuli
  Loop
    Tag Condition_1
      callMacro SET_EXPERIMENT
      EndTag
    Tag Condition_2
      callMacro SET_THERAPY
    EndTag
  EndLoop
  Loop
    Tag Mode_A
      callMacro SET_MODE_WITH_BEAM
    EndTag
    Tag Mode_B
      callMacro SET_MODE_NO_BEAM
    EndTag
  EndLoop

  ... -- Code common for all 4 interations

EndProcess
```

The previous code would result in 4 individual test stimuli executions, with small variations as seen below in Table 1.

Table 1: Expanded Loops Execution

Execution	Conditions
#1	Experiment with beam
#2	Experiment without beam
#3	Therapy with beam
#4	Therapy without beam

SYSTEM ARCHITECTURE

In order to execute the unit tests, a test stand was developed. As seen in Figure 1, it emulates every subsystem of the facility to which PaSS is connected, such as the control systems, sensors and actuators at the interface level.

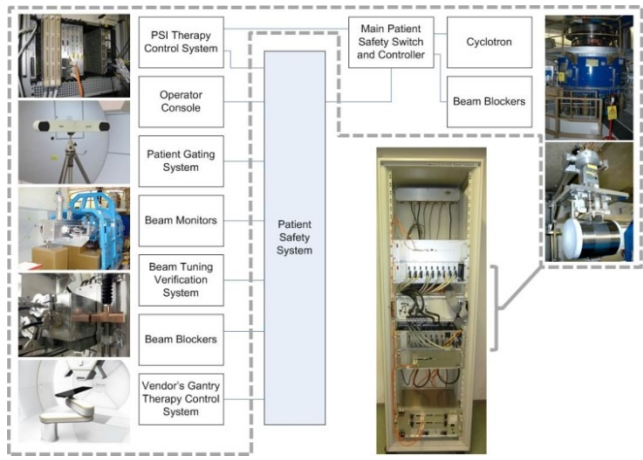

Figure 1: Patient Safety System test stand layout.

Hardware

The centre of the test stand is a National Instruments´ PXI crate with multiple IO cards. The signal lines are driven using fast PXIe-6535 cards clocked at 1MHz. Slower PXI-6509 cards were used for slow control, like defining the direction of lines and reading temperature sensors. An in-house designed 19 inch backplane board is used to route the digital IO pins to the corresponding PaSS signals. Attached to the backplane are modular carrier boards with mounted plugins that convert the 5V digital signals to and from the PXI crate into the corresponding interface of each PaSS line, see Figure 2. There are plugins for optical signals, 24 V digital lines, 5V TTL and three wire, redundant current loops in use now. More plugins could be developed to interface other types of signals.

Each PXI crate used is limited to 400 IO lines, which is enough for the currently installed PaSS systems at our facility. Should more lines be required in the future, the PXI crates allow for a daisy chain synchronization of several crates, therefore extending the available IOs to whatever might be necessary.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA098

Figure 2: Backplane and several plugin types.

Software

The application running on the PXI crate is written in LabView. It is modular and extendable, containing the following parts: The configuration module parses an xml file describing the name and type of each of the signals configures accordingly the fast IO lines, backplane and plugins. The unit test parser reads the test descriptions and creates an internal executable data structure with stimuli, assertions and time measurements. It also expands macros and tests with loops into multiple simple tests to be executed sequentially. The execution module uses the internal data structure to drive the input signals to the desired stimuli patterns and monitors the outputs. This is done synchronously for all IOs. The report module creates a final document with one section per unit test executed, and generates summaries, tables and time diagrams to facilitate the interpretation. Currently the report is written in MS Doc format.

The user interface includes a test execution GUI, as seen in Figure 3, which guides the user in loading the different configuration and test files, giving feedback of the internal structures created, syntax errors or hardware configuration issues.

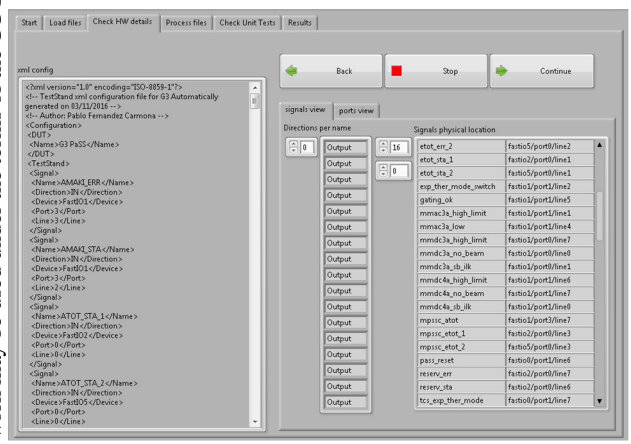

Figure 3: Sequential test execution GUI.

After the user loads all files and confirms the hardware configuration, all tests are autonomously executed and a final report is generated, ready to be checked and signed by the unit tester.

In order to facilitate debugging, a logfile viewer tool is also available, as seen in Figure 4. It has proved itself useful when debugging logic, and includes a number of filters to ease sorting the most relevant lines.

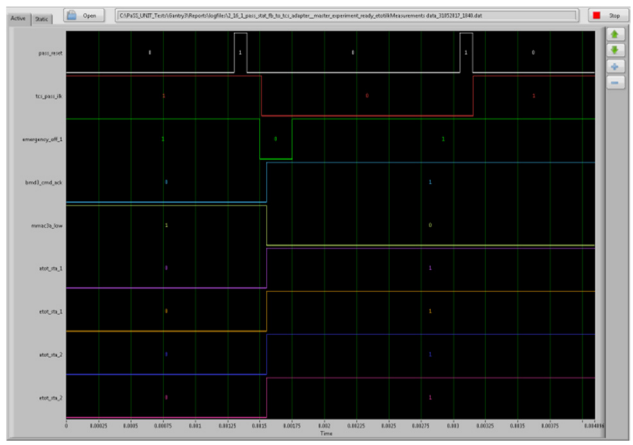

Figure 4: Logfile analysis and debug GUI.

As the application is modular, it would be possible to easily extend it to generate different report formats, or to support new unit test description languages.

REPORTING

After sequential execution of all unit tests, a report is generated automatically. It includes a list of all tested items and its success, information of failing assertions and time measurement results including a generated diagram. Figure 5 shows one section of the final report containing the results for one item.

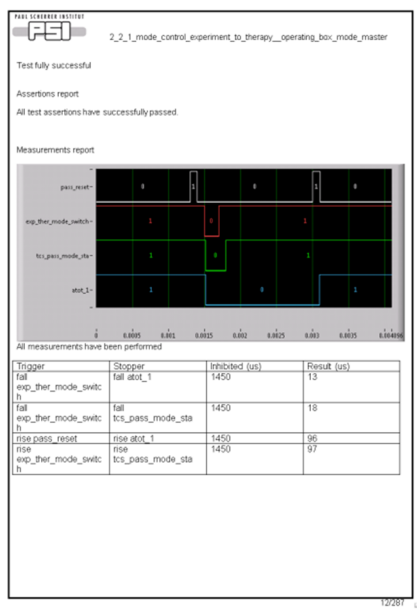

Figure 5: Extract of unit test report section.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA098

RESULTS

The report for our new Gantry 3 contains 287 pages. It includes 278 different tests defined in 13 unit test specifications. The execution of all unit tests plus the report generation takes 4 minutes.

The experience of using the new test stand for the unit testing of the PaSS of the new Gantry 3 is summarized in Table 2 below.

Table 2: Work Effort with New Test Stand

System	Effort/Capabilities
Unit test specification	3 weeks(Test description document, Formal language)
Hardware setup	½ day
PaSS unit test execution	4 minutes
Signal stimuli	≤ 400 per PXI chassis. Daisy chainable
Signal monitoring	≤ 400 per PXI chassis. Daisy chainable

As a reference, work effort is compared in Table 3 to the old testing procedure consisting of oscilloscope tests and manipulation of monitors and actuators in the facility, of the old test stand system in order to stimulate up to 40 signals at a time in the lab.

Table 3: Work Effort with Old Test Stand

System	Effort/Capabilities
Unit test specification	3 weeks(Test description document including timing diagrams)
Hardware setup	1 day
PaSS unit test execution	1 hour per test case 2 weeks in total
Signal stimuli	Manipulations in real system or ≤ 40 signals in lab
Signal monitoring	48 channels logic analyser

It can be seen that the new test stand provides not only a faster execution but also a more complete coverage of test cases. Also it allows for precise time measurements, not technically possible with the older setup.

CONCLUSION

At PSI, a test stand has been developed to automate part of the QA of the Patient Safety System of our newly installed Gantry 3. It is fast, precise and extendable. The unit tests are described in a formal language and reports are generated automatically upon execution of all test cases.

By automating the unit testing of PaSS, an increased level of safety has been achieved. It allows very complete tests scenarios and the beam time available for patients can be substantially increased, by reducing the requirements for this QA. The development cycles in upgrades and bug fixing have also been shortened, therefore reducing costs.

REFERENCES

[1] E. Pedroni, "Commissioning and use of the PSI spot scanning isocentric system for proton therapy," in *Proceedings of the 14th International Conference on Cyclotrons and their Applications*, Cape Town, South Africa, 1995.

[2] E. Pedroni, "The 200-MeV proton therapy project at the Paul Scherrer Institute: Conceptual design and practical realization," *Med. Phys.,* no. 22, pp. 37-53, 1995.

[3] Varian Medical Systems, [Online]. Available: https://www.varian.com/oncology/solutions/proton-therapy.

[4] O. Actis, F. Albertini, R. Besson, C. Bula, F. Emmert, F. Gagnon-Moisan, C. Gomà, M. Grossman, J. Hrbacek, S. König, T. Lomax, A. Mayor and S. Safai, "Gantry 2 Quality Assurance Manual", Internal report for regulatory body, Paul Scherrer Institute, Villigen, Switzerland, 2013.