Implementation of Software for RPS Verification Facility in SMART-PPE

JeongHeui Kim [a*], SeongJin Kim[a], DooHwan Kim[a], JuHong Park[a]
[a] *System Researech Institute, Hyundai Venture Ville 3F 304, 10, Bamgogae-ro 1-gil Gananm-gu, Seoul, Korea*
**Corresponding author: ganuri@soosan.co.kr*

1. Introduction

As a part of a design validation of SMART MMIS Safety System Project, We manufactured RPS verification facility for SMART-PPE. The RPS provides a shutdown of the reactor to protect the core. The RPS also provideds ESF actuation functions required to prevent the release of significant amounts of radioactive material to the environment during an abnormal release of radioactivity into the confinement air. The RPS cabinet consist of the Bistable Processor(BP), Coincident Processor(CP), Interface Test Processor (ITP), initiation circuits, and other devices necessary to monitor the selected process parameters and initiate reactor trip upon detection of non-permissible conditions. SMART COre Protection System(SCOPS) is also included in RPS cabinet. SCOPS shares MTP and ITP. We have implemented BP, CP, ITP and MTP except SCOPS.

In this paper, Implementation of software for RPS verification facility in SMART-PPE is described.

2. Configuration for RPS Verification Facility

Fig.1 shows the configuration of the RPS verification facility for SMART-PPE.

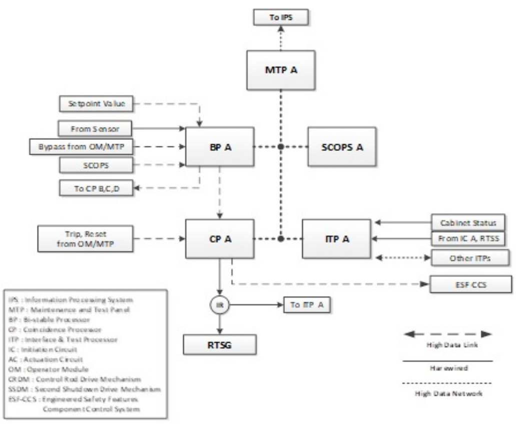

Fig. 1. Configuration of RPS Verification Facility

BP compares the input value with a fixed or variable setpoint. If any safety parameter value exceeds the setpoint, the appropriate initiation signals are generated. Setpoint is two types, one is the fixed setpoint and one is variable setpoint. BP includes not only trip output but also pretrip output for trip advance warning. In addition, BP includes functions for alarm, diagnosis, and operating bypass.

The CP provides 2-out-of-4 logic for RT and ESF actuation. The CP in each channel provides a coincidence logic algorithm per the BP trip function. The CP produces a coincidence signal for any of the following bistable trip inputs: AB, AC, AD, BD, CD, ABC, ABD, ACD, BCD, ABCD. The CP algorithm determines the state of the coincidence output based on the status of the four trip channel inputs and their respective trip channel bypass inputs. When a trip channel bypass is present, the coincidence logic provides 2-out-of-3 logic for RT and ESF actuation.

The ITP monitors the RPS status and is used to initiate manual and/or automatic surveillance testing based on operator input via the MTP.

The MTP provides functions of operating bypass, trip channel bypass, setpoint change and test initiation. it also display the setpoint, system status, test results and system status.

2.1 Design of Software for RPS Verification Facility

Software Requirement Specification(SRS) and Software Desing Specification(SDS) are written according to reference [2] and reference [3]. and then softeware for RPS verification facility is implementated. BP software is divided main program and sub program(User function). Main programs is organized BP_Algorithm and BP_Diagnosis. Sub program is user function. The user function for BP is shown in Table I.

BP_Algorithm consists of 46 parameter logic, input logic, output logic, setpoint logic and test logic. If necessary, each parameter use to call a user function.

BP_Diagnosis consists of power, NCPU_2Q, NFD2_1Q, NFD1_5Q, NFD1_6Q, NADF_1A, NI_D23Q and OUTPUT_PRO.

Table I: User Function for BP

Function	Description
CompareHigh	Compare for rising trip parameters
CompareLow	Compare for falling trip parameters
EngConvert_Lin	Convert to engineering unit(Linear)
EngConvert_Log	Convert to engineering unit(Logarithim)
Heartbeat	Heartbeat of BP
Manual_Reset_Setpoint	Manual reset setpoint for LPP and LMSP
Rate_Setpoint_Control	Rate limit setpoint for VOP
OBS_high	Operating bypass(High)
OBS_Low	Operating bypass(Low)
SDN_WRITE_REAL	Communication of SDN
SDN_WRITE_UDINT	Communication of SDN
SEL_SWITCH	Selection of parameters

CP software is divided main program and sub program(User function). Main programs is organized CP_Algorithm and CP_Diagnosis. Sub program is user function. The user function for CP is shown in Table I.

CP_Algorithm consists of 46 parameter logic, input logic, output logic, and test logic. If necessary, each parameter use to call a user function.

CP_Diagnosis consists of power, NCPU_2Q, NFD2_1Q, NFD1_5Q, NFD1_6Q, NI_D23Q and OUTPUT_PRO.

Table II: User Function for CP

Function	Description
CC_Logic	Coincidence logic for parameters
Channel_Bypass	Selection for a priority and FIFO of channel bypass among channel A, B, C, D
Initiation_Logic	Initiation of reactor trip and ESF-CCS trip
Reset_Logic	Reset of reactor trip and ESF-CCS trip
SDN_WRITE_REAL	Communication of SDN
SDN_WRITE_UDINT	Communication of SDN
Test_Enable_Control	Selection of test parameter

ITP software is divided main program and sub program(User function). Main programs is organized ITP_Algorithm and ITP_Diagnosis. Sub program is user function. The user function for ITP is shown in Table III.

ITP_Algorithm consists of INPUT_PRO, Normal_Test, Manual_Test, Auto_Test anf OUTPUT_PRO. If necessary, each parameter use to call a user function.

ITP_Diagnosis consists of power, NCPU_2Q, NFD2_1Q, NQ_D23Q NI_D23Q, NADF_1Q and OUTPUT_PRO.

Table III: User Function for ITP

Function	Description
AT_BistableCompare	Auto BP test
AT_CC	Auto CP Test
DEV_CHECK_BOOL	Normal test(BOOL)
DEV_CHECK_REAL	Normal test(REAL-RATE)
DEV_CHECK_REAL1	Normal test(REAL-FIX)
SDN_WRITE_UDINT	Communication of SDN
SDN_READ_UDINT	Communication of SDN
SEQ_CONTROL	Sequence for parameters

2.2 Design Validation

Software development method is adopted reference [5]. Especially, a modified incremental SDLC model was used according to the project schedule of SMART. The modified incremental model is a method of software development where the software is designed, implemented and tested incrementally until the software is finished. The software is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

For the design validation of RPS software, Component Test(CT), Integration Test(IT), System Test(ST) and Acceptance Test(FAT) were ferformed.

CT is done at the lowest level of software. It tests the basic unit of software such as software module of each processors such as BP, CP, and ITP.

IT is performed at the integrated units based on the information in the software design specifications.

ST tests the integrated software and hardware based on the information in the system requirements.

Finally, FAT is done when the completed system is handed over form the developers to the customers or users.

In the CT, the branch coverage test was accomplished based on white box test technique. Table IV shows the CT result for all module of BP, CP and ITP

Table IV: Test Result of CT

	Pass(%)	All Objects	Passed Objects	Failed Objects
BP				
Test Features	100%	54	54	-
Test Cases	100%	448	448	-
Coverage	100%	448	448	-
CP				
Test Features	100%	52	52	-
Test Cases	100%	738	738	-
Coverage	100%	738	738	-
ITP				
Test Features	100%	3	3	-
Test Cases	100%	24	24	-
Coverage	100%	24	24	-

In the IT, test features were divided by three sub system such as BP, CP, ITP. Table V shows the IT result for all module of BP, CP and ITP.

Table V: Test Result of IT

	Pass(%)	All Objects	Passed Objects	Failed Objects
BP				
Test Features	100%	1	1	-
Test Cases	100%	9	9	-
Coverage	100%	9	9	-
CP				
Test Features	100%	1	1	-
Test Cases	100%	4	4	-
Coverage	100%	4	4	-
ITP				
Test Features	100%	1	1	-
Test Cases	100%	9	9	-
Coverage	100%	9	9	-

In the ST, test were performed all function according to system requirements. Table VI shows the ST result.

Table VI: Test Result of ST

	Pass(%)	All Objects	Passed Objects	Failed Objects
Test Features	100%	8	8	-
Test Cases	100%	177	177	-

3. Conclusions

The implementation of software for RPS for SMART-PPE was completed according to the modified incremental SDLC model and the design validation tests were successfully accomplished through this project. All tests (CT, IT, ST) are successfully passed.

REFERENCES

[1] SeoRyong Koo, Seop Hur and ChangHwoi Kim, Development and Design Validation of Reactor Protection System Verification Facility for SMART , Transactions of the Korean Nuclear Society Spring Meeting, 2019.

[2] IEEE 830, IEEE Recommended Practice for Software Requirements Specification, 1998.

[3] IEEE Std. 1016, IEEE Standard for Information Technology-Systems Design-Software Design Descriptions, 2009.

[4] IEEE Std. 1008, IEEE Standard for Software Unit Testing, 19878.

[5] IEEE Std. 1012, IEEE Standard for Software Verification and Validation, 2004.