POINT LEPREAU REFURBISHMENT PROJECT PROGRAMMABLE DIGITAL COMPARATOR (PDC) REPLACEMENT FOR SDS1 AND SDS2 - UPDATE 2 SOFTWARE DEVELOPMENT AND REVIEW AND VALIDATION TEST RIG DEVELOPMENT

By

K.G. Fraser, M. Zhao, A. Condor, A. McDonald
Atomic Energy of Canada Ltd.
2251 Speakman Dr.
Mississauga, Ont. L5K 1B2

P.D. Thompson
NB Power Nuclear
Point Lepreau Generating Station
PO Box 600, Lepreau NB
E5J 2S6

Abstract

As part of the Point Lepreau Refurbishment Project the Programmable Digital Comparators (PDCs) for both shutdown systems are being replaced. This paper describes the progress made on the PDC replacement since the update one year ago. The project has completed the software requirements and design and these have been subject to formal reviews. At the time of presentation of this paper, the contracts will have been placed for the supply of hardware for both SDS1 and SDS2 computer systems. CNSC interaction and the planned future work are also discussed. The upfront development progress on validation tools is also presented.

1 Introduction

NB Power Nuclear is planning to conduct an 18-month maintenance outage of the Point Lepreau Generating Station (PLGS) starting in April 2008. The scope of the outage was determined from the outcome of a two-year study (Phase 1) involving a detailed condition assessment of the station which examined issues relating to ageing and obsolescence, along with a detailed review of Safety & Licensing issues associated with extended operation. The Project has proceeded with the implementation of the scope identified in Phase 1. This is referred to as Phase 2 of the Project [1].

The Phase 1 assessments concluded that replacement of the PDCs (Programmable Digital Comparators) for both shutdown systems was required in order to ensure operation of the plant for a further 25-30 years.

A paper [2] was presented at the 2003 CNS conference that discussed the method used to select the replacement hardware platform, the Tricon system, for the Shutdown System Number One (SDS1) PDCs, and the development of the software development approach that would be used for the SDS1 and SDS2 application software.

A follow-up paper [3] was presented at the 2005 CNS conference that discussed the qualification of the new PDC platform for SDS1 and the development of software engineering tools and facilities to be used in the software development.

This paper picks up where the last paper left off and discusses the activities performed up to early 2006. This includes the development of the software requirements and software design, formal reviews of these, hardware procurement activities, and development of the PDC validation tools.

Finally, the remaining work that will be performed on the PDCs during Phase 2 of the Project is discussed.

2 The Programmable Digital Comparators

Two completely independent shutdown systems, SDS1 and SDS2, are used in CANDU (CANada Deuterium Uranium) reactors.

Each system contains three independent safety channels arranged in a 2-out-of-3 voting system. Channelized instrumentation is used to monitor a number of plant neutronic and process variables. If variables in any two channels of a single system are outside pre-determined envelopes, a shutdown is initiated.

At PLGS the logic for most of the process-related reactor trip coverage is implemented in computers. This allows optimization of trip functions for various operating conditions and reduces the burden of calibration, testing or maintenance activities that would otherwise have been required with conventional analog and relay-based logic. A denser packaging of logic is made possible, with the corresponding saving of space, while increasing the flexibility in implementing enhanced trip functionality. Computers also allow implementation of self-checking and equipment monitoring functions to improve availability and decrease the maintenance and testing work load on the station staff.

The computers are commonly referred as Programmable Digital Comparators to emphasize the relatively simple computing functions contained within the computers. The configuration

introduced in the CANDU 6 reactors was two independent PDCs per channel, for a total of six PDCs per shutdown system.

The original PDC was a general-purpose mini-computer from Data General, the MP 100. This computer was very well established in process control applications, particularly in the pulp and paper industry.

For newer plants, such as Wolsong 2,3,4, Qinshan 1, 2, and Cernavoda 2, the original PDC equipment was no longer available from the manufacturer. Discussions with the manufacturer revealed that some integrated circuits used in the PDCs were no longer in production. This meant that the manufacturer was not able to get the parts needed to produce the PDCs. As the manufacturer no longer had a suitable product line, new hardware suppliers had to be selected.

For SDS1 the selected computer hardware was from ABB (Asea Brown Boveri) and the product line was called Procontrol P13. For SDS2, the computer hardware was supplied by PEP Modular Computers (now Kontron). It uses an industry standard VME based architecture.

3 Point Lepreau PDCs Platform Selection and Qualification

Condition assessments were carried out in 2000-2001, to assess plant equipment and systems that should be replaced as part of the refurbishment. It was concluded that replacement of the PDCs for both shutdown systems was required to ensure operation of the plant for a further 25-30 years.

The recommendation from Phase 1 for SDS2 was to use the same hardware and software development methodologies as had been used for Wolsong 2,3,4, Qinshan 1 and 2, and Cernavoda 2. The hardware platform is considered resistant to obsolescence and it was concluded that the hardware would still be maintainable for the remainder of the station life.

For SDS1, the recommendation from Phase 1 was to use a different hardware platform since it was considered not likely that the ABB hardware could be maintained for the remainder of the station life. The hardware platform must support a graphical function block software language approach similar to that used on SDS1 for Wolsong 2,3,4 and later projects. The graphical approach is favoured because it has proven effective in the past at simplifying the software engineering process.

An assessment was performed [2] against the various requirements and the Tricon platform from Invensys Triconex was selected. This platform has been qualified for use in the shutdown system [3] and the qualification report has been accepted by the CNSC.

4 Design Independence

Design independence ensures that the chance of making mistakes common to development, verification and validation is minimized and that the verification and validation processes can be performed effectively and objectively.

Within each shutdown system, three independent teams are defined: developer, verifier, and validator. Each team has its own supervisor and the management of the Validation and Reliability Qualification Team is separate from that of the Development and Verification Teams. In addition, the tools and facilities used by the Verification and Validation teams are independent.

The following design independence is enforced between SDS1 and SDS2 teams:

a) No member of the project shall participate in the developer role for both SDS1 and SDS2.
b) No member of the Software Development Team in one system shall act as a verifier for either system.

5 Software Lifecycle

Figure 1 shows the software lifecycle for the PDCs.

Software development for the PDCs consists of three phases, the software requirements specification (SRS) which is derived from the design input documentation (DID), the software design description (SDD), and coding.

The verification activities consist of review, verification, and testing. Each phase is reviewed against the previous phase to ensure compliance. The SDD and the code are mathematically verified against the SRS and SDD respectively. Unit and subsystem testing are used to test the SDD and the code against the previous phases.

The software hazards analysis is used to identify hazards in the software design and code. This process can lead to changes that make the software more robust and resistant to failures.

Finally, validation and reliability qualification testing check the code against the DID requirements.

6 Software Development

6.1 Software Requirements Specification

Specification of the software requirements has been completed for both shutdown systems, although further revisions are anticipated to cover system requirements changes to further enhance safety. The software requirements for the PDC software are defined in terms of monitored and controlled variables. In the SRS, the PDC is treated as a black box with electrical signals entering and exiting the system. The software functional requirements, defining the relationship between the monitored and controlled variables, are specified using a formally defined notation with unambiguous mathematical definitions. Different approaches are used for SDS1 and SDS2. Other requirements, such as performance requirements and accuracy requirements are specified separately.

SDS1 uses the Integrated Approach (IA), which is a formal methodology developed by AECL for the design and verification of safety-critical software. It proceeds from the design input documentation through the entire software development cycle. A major characteristic of the IA is the use of a mathematically precise function block language to specify the software requirements. The function block language used for PLR is defined in the IEC 61131-3 standard.

Figure 2 shows a sample of software requirements specified using the integrated approach. Input signals are shown on the left and outputs on the right. The logic is in between. Each block is mathematically defined and the diagrams can be nested to allow re-use of common code.

SDS2 uses an approach called the Rational Design Process (RDP). The functional requirements are specified using mathematical expressions in a tabular format that identifies the relationships between variables. Figure 3 shows a sample of software requirements specified using a structured decision table following the RDP approach.

Both shutdown systems have under-gone two revisions of the SRS documents and findings from the SRS review (see Section 6.1) have been incorporated.

6.2 Software Design Description

The SDS1 and SDS2 software design descriptions have been completed; although further revisions are anticipated to cover system requirements changes.

Using the Integrated Approach, the SDS1 software design is nearly identical to the requirements. The major difference is that the requirements show the relationship between monitored and controlled variables (in electrical units), whereas the software design shows the relationship between software input and output variables.

The software input and output variables are the values read and written by the software. For example, in the case of the Tricon platform these are in the form of integer counts between 0 and 4095. This represents a 12-bit range. In order to minimize differences between the SRS and SDD, these values in counts are converted back to millivolts in software. This allows calculations to be made using floating-point numbers rather than integers and eliminates the need to recalculate all the setpoints and other constants in counts. Generally, the function block diagrams are unchanged between SRS and SDD. This also simplifies verification activities.

In addition, with the Tricon system, code is generated automatically from the SDD, which eliminates the manual coding stage.

The SDS2 SDD is done in two stages. The first stage is called the module decomposition and shows the breakdown of the software into modules. The structure of the modules is also shown. At this stage the SDD undergoes a technical review with other developers to evaluate the module decomposition.

In the second stage, the detailed design of each module is described. The design is specified in a tabular format that is similar to the SRS. For comparison, Figure 4 shows the design of the SRS logic that was shown in Figure 3.

7 Verification

7.1 Software Requirements Review

The purpose of the software requirements review is to ensure that the SRS meets all the system level requirements relevant to the software. The review also ensures that the SRS was created in compliance with the SRS procedure. The review has been completed for both shutdown systems. Any changes made to the SRS from this point on will also be reviewed.

Independent review teams performed the reviews for each shutdown system. The review teams are led by a member of the verification team and consist of independent reviewers from the validation team, the verification team, and the hardware team. Comments are resolved with the participation of the SRS author.

The major part of the review is to perform a requirements mapping between the SRS and the DID. This ensures that all relevant requirements from the DID are correctly described in the SRS and that any new requirements are justified. Figure 5 shows an example cross-reference table that shows the mapping between DID and SRS requirements.

For SDS1 the review involves comparing the function block diagrams in the SRS with the requirements in the DID. Likewise, the review on SDS2 focuses on the tabular logic used in the rational design process.

Both shutdown systems have undergone two rounds of review, with the findings from both rounds incorporated into the latest versions of the software requirements specifications.

7.2 Software Design Review

The purpose of the design review is to ensure that the software requirements are implemented appropriately in the design using the software design procedures. It also ensures that any new functionality in the design is justified.

Independent review teams perform the reviews for each shutdown system. The review teams are led by a member of the verification team and consist of independent reviewers from the verification team and the hardware team. For SDS2, another member of the verification team checks the results of each verifier. This additional step is necessary due to the more complex nature of the SDS2 process. For SDS1, there are few differences between the SRS and SDD and additional review of the verifier's work is not necessary. Comments are resolved with the participation of the SDD author.

Both shutdown systems have undergone review and the findings have been incorporated into the software design descriptions. Any changes to the design will be reviewed in the same manner.

7.3 Hazards Analysis

The Hazards Analysis identifies failure modes associated with the software. The Hazards Analysis consists of three phases: the System Hazards Assessment, the Software Design Hazards Assessment, and the Code Hazards Assessment. To date the system and software design hazards assessments have been completed for both shutdown systems. The code hazards assessment is in progress.

The System Hazards Assessment uses a Failure Modes and Effects Analysis to identify system level hazards for SDS1. From these hazards, the software related hazards are identified. The Software Design Hazards Assessment examines the software design contained in the SDD for any associated failure modes. The Code Hazards Assessment will involve an extension of the design analysis to identify failure modes that may be introduced in the code. Findings from the latter two phases are incorporated into the software design or code listings.

Figure 6 shows a sample from the system hazards assessment.

The findings from the software design hazards assessments have been incorporated into the SDS1 and SDS2 software design documents.

8 Hardware Procurement

The tendering documents were prepared and sent to the suppliers for bids for both shutdown systems in early 2006. The bids were received and evaluated and contracts awarded in spring of 2006. Triconex was selected to supply hardware for SDS1 and Kontron was selected to supply hardware for SDS2. The suppliers are currently manufacturing the PDC hardware with delivery expected this year (2006).

9 Validation Test Rig Development

The Validation test rigs used in previous projects were developed by AECL in the early 1980s. Based on various computer platforms, they were successfully used in:

- Validation testing for the PDCs currently in use in CANDU 6 stations, including Point Lepreau.
- Validation testing of Darlington trip computers.
- Validation and Reliability Qualification (V&R) Testing of SDS1 and SDS2 PDCs for Wolsong 2, 3 & 4.
- Validation Testing of SDS1 and SDS2 PDCs for Qinshan 1&2, Cernovada 2, and the replacement PDCs for Wolsong 1.

The test rig hardware used for the most recent projects (Wolsong, Qinshan and Cernavoda) was composed of the National Instruments (NI) Data Acquisition (DAQ) cards made in early 1990s. Since then, there has been advancement in the NI technology. The replacement of these DAQ cards is difficult, due to obsolescence.

The test rig software, which is referred to as ATLIN (AECL Test Language Interpreter), for these recent projects was based on an older version of the NI LabVIEW. The newer version of LabVIEW does not work with these legacy NI DAQ cards. Consequently there is a requirement to update the software to be compatible with the new generation of NI DAQ cards.

In addition, the selected Tricon-based architecture for the SDS1 PDC has doubled the input/output requirements for the Validation test rig because of the single PDC per channel configuration. Previously, the rig could only be configured for one PDC (i.e. half a channel) at a time.

Therefore, it is necessary to build new Validation test rigs with updated software and hardware for PLR.

The NI platform was selected again for the new Validation test rigs to minimize the risk and the impact on the existing Validation and Reliability test cases (scripts), thus reducing the effort required to update these test scripts.

The new Validation test rig hardware will be composed of NI PXI (PCI Extensions for Instrumentation) system. The NI PXI system was selected for the following reasons:

- Synchronization between all DAQ cards inserted into the PXI chassis is inherent. No additional means are required.
- Timing measurements are separate from the main software routine without using software polling and are independent from the operating system. The measurement data are stored in the on-board buffers of the DAQ cards. This significantly increases the accuracy of the timing measurements to an anticipated better than 2 ms and provides deterministic performance.

The new ATLIN will be based on NI LabVIEW 8. Use of built-in functions in the newer version of LabVIEW makes the test rig software (ATLIN) readable, scalable, and more easily maintainable.

For the PLR Project, additional test hardware has been provided to emulate (simulate) the PDC functions, providing functionality not available on previous test rigs. This PDC Simulator design is based on the design requirements for the actual PDCs. It is implemented in a separate NI PXI system, running in real-time.

The function of this PDC Simulator is to allow extensive testing of ATLIN integrated with the NI hardware, in advance of receipt of the actual PDC application software.

In addition, all V&R preparation work on the test procedures and test scripts can be advanced using the PDC Simulator, which reduces the schedule risk by bringing forward the end date required to do this preparation work, prior to receipt of the PDC and the application software.

Furthermore, the PDC Simulator is being considered as a means to generate PDC predicted responses that can be compared with the actual PDC responses, in the Reliability Qualification Testing.

10 CNSC Interaction

In March 2005, the comments received on the last pre-production document were resolved to the CNSC's satisfaction. This completed CNSC acceptance of all pre-production documents.

Since the CNSC has accepted all of the software development procedures, it is not necessary for them to review in detail all of the design documentation as it is produced. Rather, the intent for the production phase of the work is to submit to the CNSC the verification and validation documents, to provide objective evidence that the accepted process is being followed.

To date, the software requirements review documents for both SDS1 and SDS2 have been submitted to the CNSC.

11 Future work

Coding for SDS2 is in progress. The verification activities will continue with the systematic design verification, code review and verification, and unit and subsystem testing. The software production and verification will be performed on the software development test rigs at AECL. These are SDS-specific, i.e. one rig for each of SDS1 and SDS2 and include the target PDC hardware (one PDC channel). In addition, the software development test rigs will contain hardware to apply the test cases for unit and subsystem testing. A second development and verification cycle is planned to correct any deficiencies identified in the first round of verification.

Upcoming work on the validation test rigs will consist of the following:

- ATLIN and the PDC Simulator may need to be updated to include any upstream changes on the functional requirements for the PDCs.
- Documentation of the new validation test rig hardware and software will be prepared.
- Development of the PDC Simulator, so that the formal V&R test procedures and test scripts will be ready before the actual PDC hardware and the application software are available.

Following completion of the validation test rigs, Validation Testing and Reliability Qualification will be conducted on both shutdown systems. The validation testing and reliability qualification are completely independent from the verification testing and use separate personnel and test equipment.

The major milestones to complete the Project are:

- Procure and install the balance of the test rig hardware
- Software Systematic Design Verification
- Completion of coding (SDS2)
- Systematic Code Verification (SDS2)
- Software Unit and Integration Testing
- Delivery of production PDCs to AECL
- Validation Testing and Reliability Qualification
- Delivery to site

12 References

[1] G. Thomas, et. al., "Point Lepreau Refurbishment - Update 6," paper to be presented at the 27th Annual conference of the Canadian Nuclear Society, June 11 - 14, 2006, Toronto, Ontario.

[2] N.M. Ichiyen, et. al., "Point Lepreau Refurbishment Project Programmable Digital Comparator (PDC) Replacement for SDS1 and SDS2," paper presented at the 24th Annual conference of the Canadian Nuclear Society, held in Toronto in June 2003.

[3] K.G. Fraser, et. al., "Point Lepreau Refurbishment Project Programmable Digital Comparator (PDC) Replacement for SDS1 and SDS2 – Update 1," paper presented at the 26th Annual conference of the Canadian Nuclear Society, held in Toronto in June 2005.

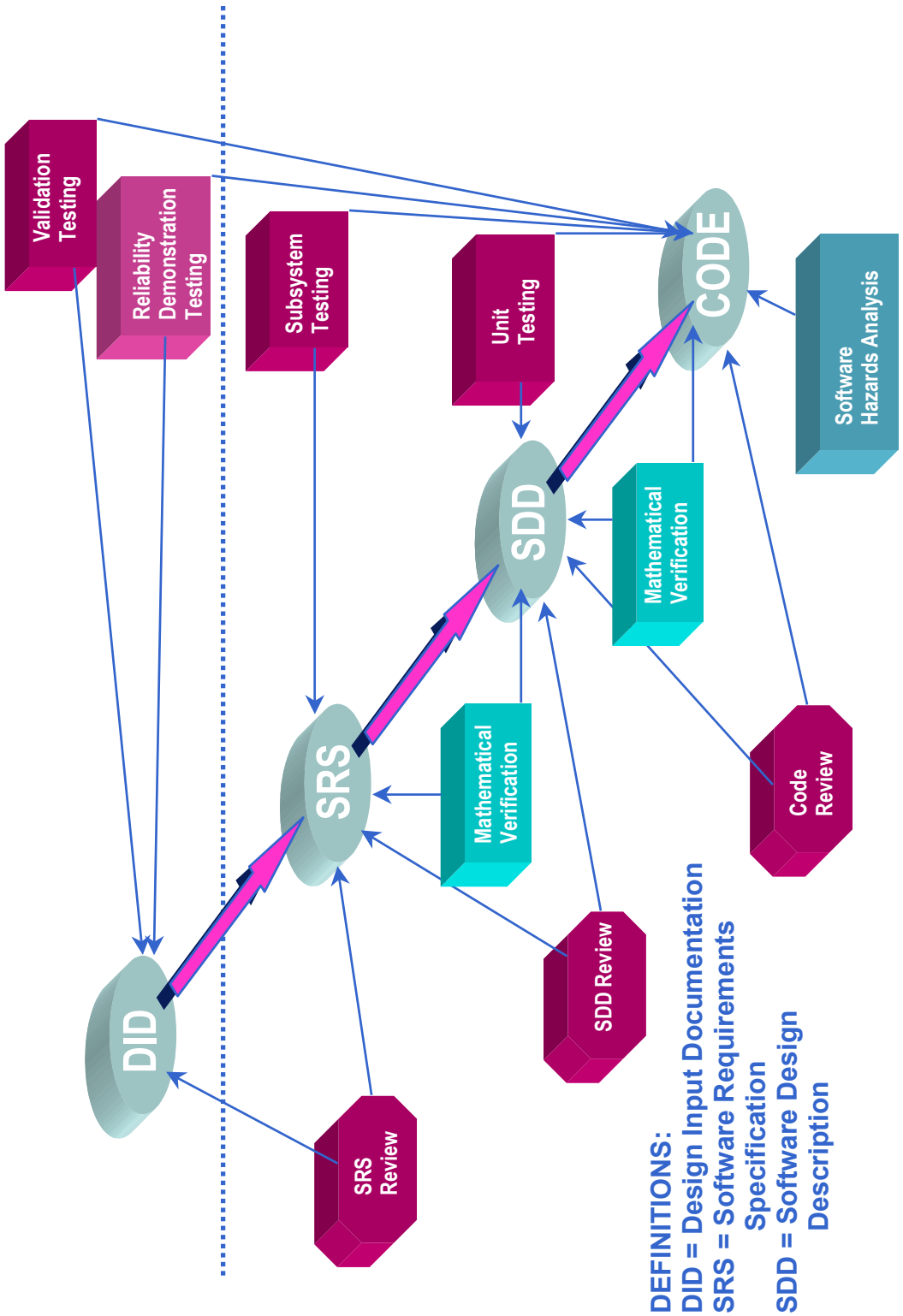

Figure 1 - Software Lifecycle

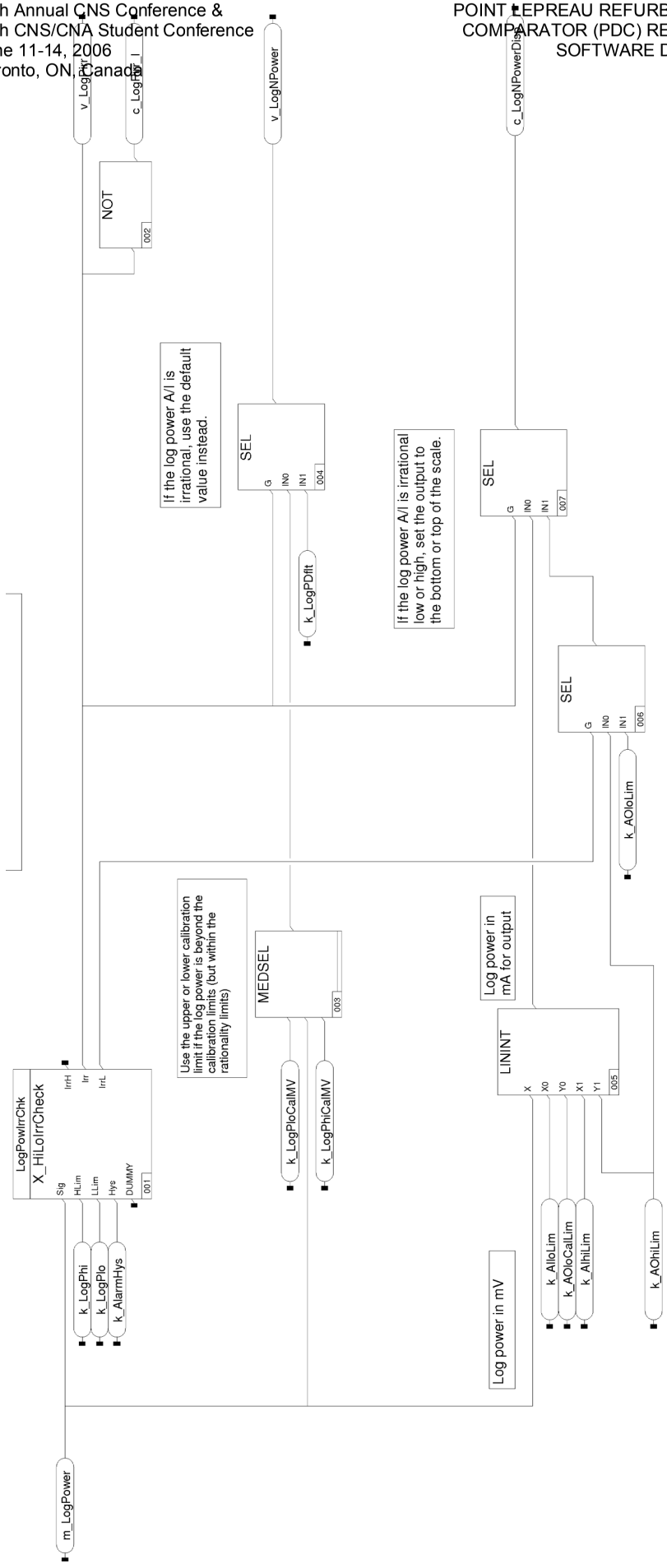

Figure 2- Example of Software Requirements using the Integrated Approach

5.1.2.13.7 f_FlogM

Determine the state of ion chamber log power signal rationality check abnormal error message.

Inputs:

Name	Data Type	Units	Reference
f_FlogAlm	t_Alm	N/A	5.1.2.13.6
m_WDogTst	t_Pb	N/A	4.1.1.16

Outputs:

Name	Description	Data Type	Units
f_FlogM	Current state of ion chamber log power signal rationality check abnormal error message.	t_Msg	N/A

SDT: f_FlogM

Condition Statements	1	2	3
m_WDogTst = e_Pressed	T	F	F
f_FlogAlm <> e_AlmNorm	-	T	F
Action Statements			
f_FlogM = e_MsgOn	X	X	
f_FlogM = e_MsgOff			X

Figure 3 - Example of Software Requirements using the Rational Design Process

Program: **EFlogM**

Parameter	*Type*	*In/Out*
(None)	--	--
Return (None)	--	--

Inputs:

Name	*Ext_Value*	*Type*	*Origin*
!MErr!	GFlogErr	<BOOLEAN>	||SnrAlmDet||
!WTst!	GmWDogTst	<WDogButton>	||ChkIO||

Outputs:

Name	*Ext_Value*	*Type*	*Origin*
--	ScFlogM(!MOut!)	<MsgStat>	||MsgOut||

Updates:

Name	*Ext_Value*	*Type*	*Origin*
(None)	--	--	--

PFT: 2.1.6.1

	!WTst! = ##WDPressed##	~(!WTst! = ##WDPressed##)	
		!MErr! = FALSE	~(!MErr! = FALSE)
!MOut!	##MsgOn##	##MsgOff##	##MsgOn##

Figure 4 - Example of Software Design using the Rational Design Process

2.4.9.3 FN-BLL.condout: Boiler Low Level Trip Conditioning Logic

	v_LogNPower < (k_BLLlogPcondSP – k_CondHys) AND PU_to_EU(v_AvgCPowFP) < (k_BLLavgPcondSP – k_CondHys)	v_LogNPower ≥ k_BLLlogPcondSP OR PU_to_EU(v_AvgCPowFP) ≥ k_BLLavgPcondSP	else
v_BLLCondOut	true	false	NC
c_BLLCondOut	true	false	NC
SRS Reference	D-12 (8-11, 13)		

Figure 5 - Example Software Requirements Review Cross-Reference

Ident. No.	Function	Failure Mode			Failure Cause(s)	Failure Detection Method	Compensating Provisions	Severity Class (Type of Failure)	Failure Effect
10.1	Trip MODLL parameter on low Moderator level trip condition within the required time	Failure to open trip D/O 34 on Moderator Low Level trip condition within the required time	Failure of PDC input signal(s)	A/I signal fails high, "as is", or falls slower than Design/ Analysis.	Hardware failure in instrumentation loop, calibration facilities and human error	SRST, PDC spread check, Routine shift panel check	Channelization, operating procedures, operator training	• Error of the same calibration facility across channels: SC I • Other hardware faults: SC II • Operator error: SC I	MODLL parameter fails to trip the channel in the required time
				LogN power signal fails low, such that the MODLL trip is incorrectly conditioned out.	Hardware failure in instrumentation loop, calibration facilities and human error	SRST, Routine shift panel check	Channelization		
			PDC Hardware Failure	Failure to open parameter trip D/O	CPU and/or D/O module failure	SRST, PDC selfchecks, watchdog	Channelization	• SC II	
				Failure to provide valid A/I signal to the CPU	A/I module failure, ADC hardware failure, watchdog failure				
			PDC Software Failure	Value of moderator level signal fails high or "as is"	Software fault, CPU failure, RAM/ROM failure, ADC s/w failure	SRST, watchdog, PDC selfchecks	Channelization (hardware faults), watchdog circuit, trip computer self checks, defensive programming techniques	• Software fault: SC I • Hardware faults: SC II	
				Value of moderator low level setpoint fails low	Software fault, CPU failure, RAM/ROM failure				
				Comparator logic for level signal(s) to MODLL setpoint fails to generate MODLL parameter trip.					
				Conditioning logic fails resulting in conditioning out the parameter trip (power conversion module, conditioning logic module)					

Figure 6 - Example System Hazards Assessment