Proceedings of Korean Nuclear Society Spring Meeting Cheju, Korea, May 2001

SOFTWARE VERIFICATION AND VALIDATION METHODOLOGY FOR ADVANCED DIGITAL REACTOR PROTECTION SYSTEM USING DIVERSE DUAL PROCESSORS TO PREVENT COMMON MODE FAILURE

Ki Chang Son, Hyun Kook Shin, Nam Hoon Lee, Seung Min Baek, and Hang Bae Kim

Korea Power Engineering Company (KOPEC)
150 Duckjin-Dong, Yusong-Ku, Taejon, KOREA 305-353

kcson@kopec.co.kr, hkshin@kopec.co.kr, lnh@kopec.co.kr,
smbaek@kopec.co.kr, hbkim@kopec.co.kr

Abstract

The Advanced Digital Reactor Protection System (ADRPS) with diverse dual processors is being developed by the National Research Lab of KOPEC for ADRPS development. One of the ADRPS goals is to develop digital Plant Protection System (PPS) free of Common Mode Failure (CMF). To prevent CMF, the principle of diversity is applied to both hardware design and software design. For the hardware diversity, two different types of CPUs are used for Bistable Processor and Local Coincidence Logic Processor. The VME based Single Board Computers (SBC) are used for the CPU hardware platforms. The QNX Operating System (OS) and the VxWorks OS are used for software diversity. Rigorous Software Verification and Validation (V&V) is also required to prevent CMF. In this paper, software V&V methodology for the ADRPS is described to enhance the ADRPS software reliability and to assure high quality of the ADRPS software[5].

Keywords: Advanced Digital Reactor Protection System, Common Mode Failure, Diversity, Software Verification and Validation, Reliability

1. Introduction

Digital systems are vulnerable to Common Mode Failure (CMF) caused by software error, which overrides the hardware redundancy. In this development, the principle of diversity, such as hardware diversity and software diversity, is considered. The hardware diversity can be achieved using two or more components with a different internal design but performing the same function. The software diversity can be achieved by implementing the ADRPS using two or more OSs which provide different OS environments[8].

There is a possibility that the designers of the ADRPS will make errors and that the errors will result in undetected faults in the ADRPS. The ADRPS may then be installed with faults in it and will cause the CMF. This could clearly result in dangerous failures. The failures that can result from these faults make it necessary to apply active measures in a systematic manner to find and correct errors. Rigorous software Verification and Validation (V&V) is also required to prevent the CMF.

The ADRPS software V&V processes provide assurance that the specified functional and reliability requirements of the system are met. To be effective, software V&V for the ADRPS should be carried out in conjunction with a well organized software development

processes. To be successful, the software V&V processes for the ADRPS must fulfil V&V requirements. This paper requires an orderly, planned approach to software V&V. Successful software V&V for the ADRPS will result in the reduction of the number of software faults that lead to the CMF.

2. System Description

The functions of ADRPS are to protect the nuclear fuel design limits and reactor coolant system boundary by tripping the reactor when the plant parameter exceeds these limits. Reactor trip is provided through an interface with the reactor trip switchgear system. The ADRPS also provides assistance in mitigating the consequences of accidents through actuation of separate Engineered Safety Features (ESF) System.

The ADRPS consists of four redundant channels (A, B, C and D as depicted on Figure 1) to satisfy single failure criteria and to improve plant availability. Each channel is comprised of Bistable Processor (BP), Local Coincidence Logic (LCL) processor, System Interface Processor (SIP), initiation logic and associated Maintenance/Test Panel (MTP). The system includes four redundant Operator Modules located on the Main Control Board, one for each channel[8].

There are two BPs per channel. Each BP receives analog inputs from the separate measurement of process sensors in each channel, and discrete signals from the Core Protection Calculator System (CPCS). The bistable function determines the trip state by comparing the measured process variable to the predetermined setpoint value.

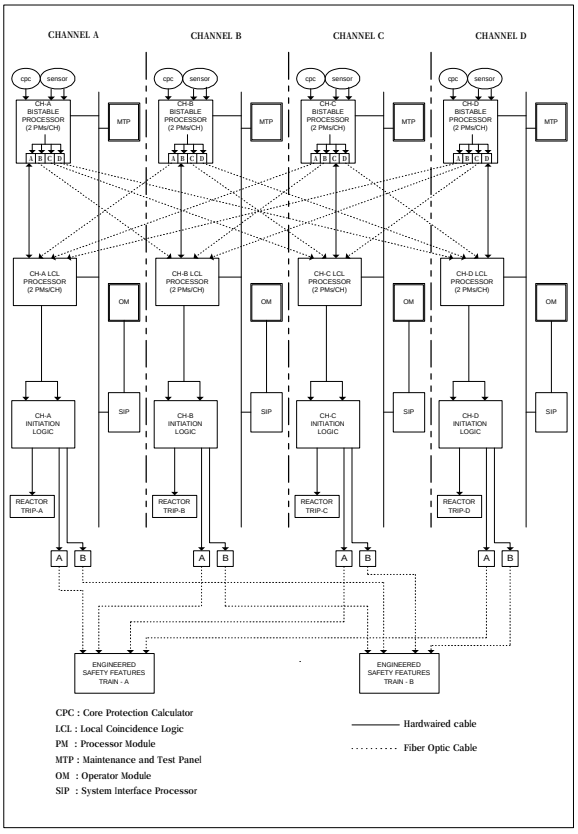

Figure 1. The ADRPS Block Diagram

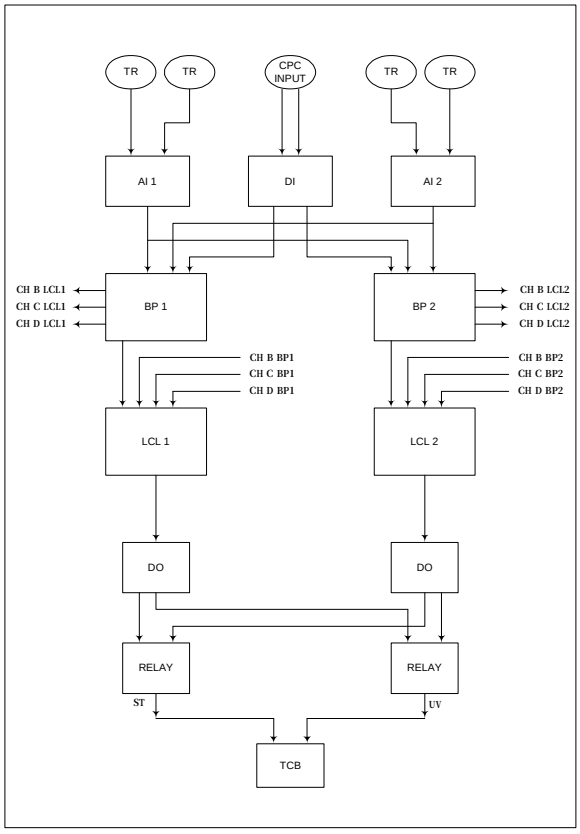

Figure 2. Single Channel Functional Block Diagram for the ADRPS

Each BP module sends its trip status to the redundant channel LCL processor modules via Serial Data Links (SDL). Each SDL utilizes fiber optic modems and fiber optic cables to provide electrical isolation between the redundant channels.

The LCL Processor includes two processor modules each of which perform two-out-of four coincidences trip logic for RT and ESFAS initiation logic. The LCL Processor produces a trip signal if two or more of four inputs indicate trip state. If a trip bypass is present, the LCL logic is converted to 2 out of 3 coincidence, from the 2 out of 4 coincidence.

The initiation signals from two LCLs are provided to the channel's dedicated initiation relays through a hardwired "OR" logic, respectively.

The MTP is provided for the primary local the ADRPS human-machine interfaces. It is used to monitor the ADRPS status, and to perform the ADRPS control functions such as the insertion of bistable trip channel bypasses and operating bypasses.

The SIP monitors the ADRPS channel status and initiates manual and/or automatic surveillance test(s) based on user input via the MTP. It provides the interfaces to external systems for status indication and surveillance testing.

3. Software Verification and Validation for the ADRPS

The software V&V for the ADRPS development is to help the development organization build quality into the software during the software life cycle[3]. The software V&V processes determine if development products of a given activity conform to the requirements of that activity, and if the software satisfies the intended use and user needs. The determination includes assessment, analysis, evaluation, review, inspection, and testing of software products and processes. The software V&V for the ADRPS development is performed in parallel with the software development for the ADRPS, not at the conclusion of the software development[2]. This report recommends software V&V methods for the ADRPS.

As shown in Figure 3, the activities of verification for the ADRPS have an essential part in the overall software life cycle of the ADRPS. The verification process is closely tied to the individual steps in the software development for the ADRPS. Verification is defined to be the check at each stage of the software life cycle to determine that there has been a faithful translation of that stage into the next one.

After the Functional Requirements Specification (FRS) for the ADRPS has been established, software verification team must check the adequacy of this in fulfilling the system requirements assigned to the software in the system design documents. After the design phase, verification addresses the adequacy of the software design in meeting the Software Requirements Specification (SRS). Likewise, after coding verification checks the compliance of the coded software with the design derived in the Software Design Description (SDD).

Verification activities were carefully planned before starting the ADRPS development project and the V&V Plan for the ADRPS constituted the most relevant section of the Software Quality Assurance Plan (SQAP) of Software Development Manual which was developed by KOPEC[7].

Verification executed phase by phase must assure that the SRS for the ADRPS is implemented in the design expressed in the SDD for the ADRPS and further into the code.

The overall verification process determines that the software behaves in accordance with its FRS.

Validation is defined to be the process of determining the level of conformance between the system requirements and the software system under operational conditions. Validation will assure that the code, when integrated with hardware and executed in the

Validation will assure that the code, when integrated with hardware and executed in the system environment, meets the requirements expressed in the SRS. However, the ADRPS can still exhibit unsatisfactory performance due to poor, incomplete or erroneous specifications

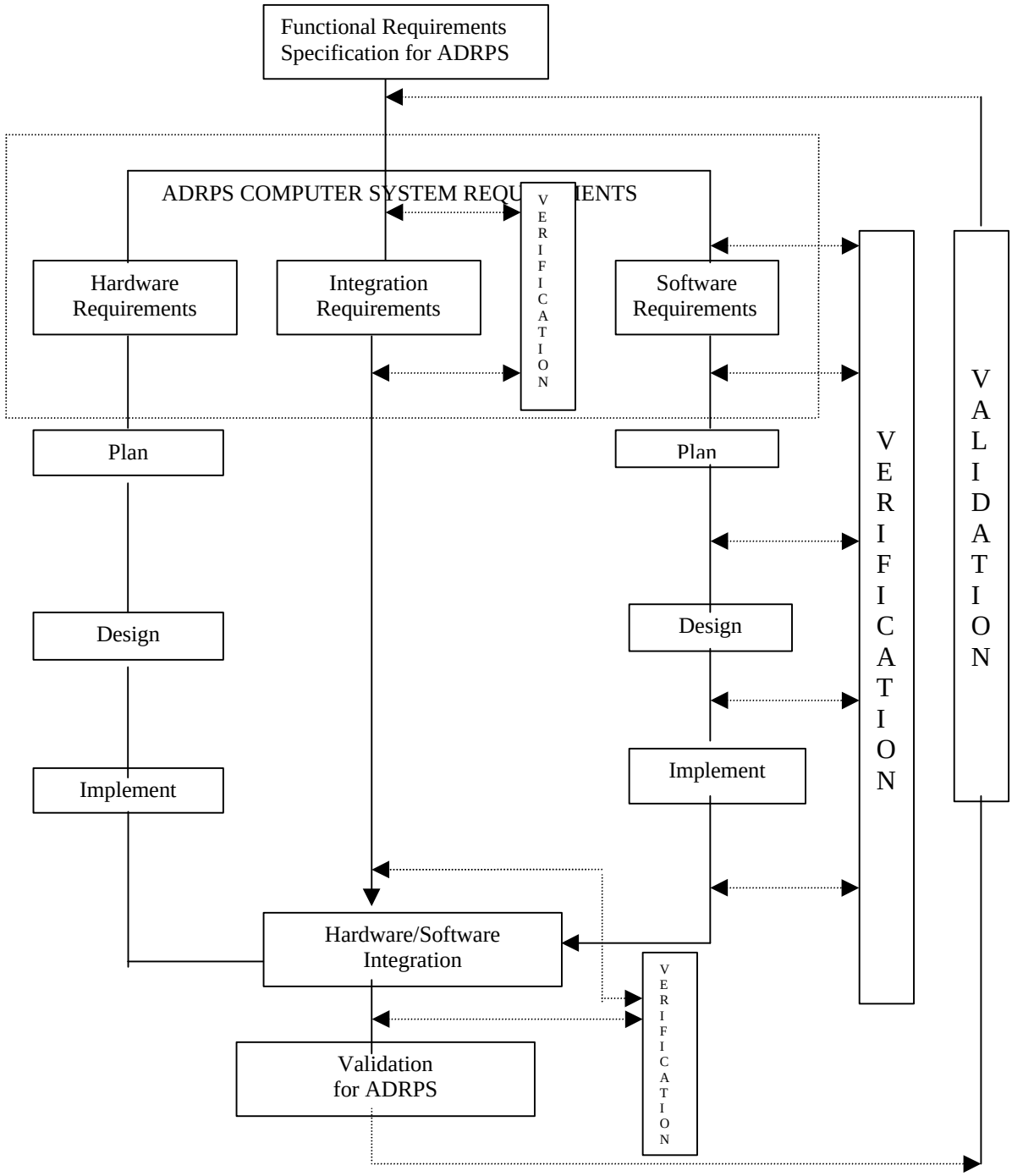

Figure 3. Development, Verification & Validation Activities for the ADRPS

3.1 Regulatory Background

The requirements for software verification and validation plan for the ADRPS are defined in ASME NQA-1-1994, Subpart 2.7, "Quality Assurance Requirements of Computer Software for Nuclear Facility Applications" and IEEE Std. 7-4.3.2-1993, "Standard Criteria for Digital Computer Systems of Nuclear Power Generating Stations". IEEE Std. 7-4.3.2-1993 has been endorsed by Regulatory Guide 1.152, Rev.01, "Criteria for Digital Computer in Safety Systems of Nuclear Power Plants". ASME NQA-1-1994 is based on software life cycle model similar to IEEE Std. 1012 which illustrates a systematic approach to software development and maintenance[6].

The development phases for the ADRPS included in a comprehensive system design process are as follows[7]:

1) Conceptual/Planning Phase

2) Requirements Phase

3) Design Phase

4) Implementation Phase

5) Integration and Test Phase

Software verification and validation for the ADRPS is performed by individuals organizationally independent of those who designed the system. The technical qualifications of the individual performing verification and validation tasks shall be comparable to those of the designers. Communications between the design team and software verification team should be documented in written reports.

3.2 Software Classifications

The ADRPS software is classified so that software engineering practices that are necessary to achieve an appropriate level of assurance for the software can be selected. All software to be used in the ADRPS are classified as one of the following classes (Table1)[7]:

SOFTWARE CLASS	DESCRIPTIONS
Safety Critical (Protection)	Software whose function is necessary to directly perform RPS control actions, ESFAS control actions, and safe shutdown control actions.
Important to Safety (ITS)	Software whose function is necessary to directly perform alternate protection system control actions or software that is relied on to monitor or test protection functions, or software that monitors plant critical safety functions.
General Purpose	Software that performs some functions other than those described in the previous classifications. This software includes tools that are used to develop software in the other classifications, but is not installed in the on-line plant system.

Table 1. Software Classifications Applied to the ADRPS

Specific parts of the software in the ADRPS may be assigned to different classes. However, each part of the software must have an assigned class. Each group responsible for developing the ADRPS software shall document the appropriate class for each software module in its software design requirements and design description. Software V&V for the ADRPS is performed and documents and materials that are produced according to the software classification of each software item. The software classifications for the ADRPS are shown in Table 2 and software tasks for each class of the ADRPS software are shown in Table 3.

SYSTEM	SUB-SYSTEM	SOFTWARE CLASS
ADRPS	Safety Critical Kernel (LCL, CCC, Bistable)	Safety Critical
	MTP Test Processor	ITS
	ITP Test Processor	ITS
	Inter-System Communication Software	ITS
	All Other Software	General Purpose
	ADRPS Software Development Tool	General Purpose

Table 2. The Software Classifications for the ADRPS

TASKS	PROTECTION	IMPORT-ANT TO SAFETY	GENERAL PURPOSE
SOFTWARE REQUIREMENTS PHASE			
Functional Requirements Spec.	X	X	
Software Requirements Spec.	X	X	
SOFTWARE DESIGN PHASE			
Software Design Description	X	X	
SOFTWARE IMPLEMENTATION PHASE			
Coding	X	X	
Test Plan	X	X	X
Unit Test Procedure	X	X	
TESTING PHASE			
Validation Test Procedure	X	X	X
Reliability Test Procedure	X	X	

Table 3. Software Tasks for Software Classifications for the ADRPS

3.3 Requirements Traceability Matrix for the ADRPS

Requirements Traceability Matrix (RTM) for the ADRPS includes all functional, hardware, software, performance, interface, and user requirements[7]. The RTM shall be a living document to be used throughout each phase of the V&V processes. The RTM is kept in a database format for ease of update using Requisite Pro which is a commercial tool. Verifier shall revise the RTM throughout software development. The example of the RTM is shown in Figure 4 .

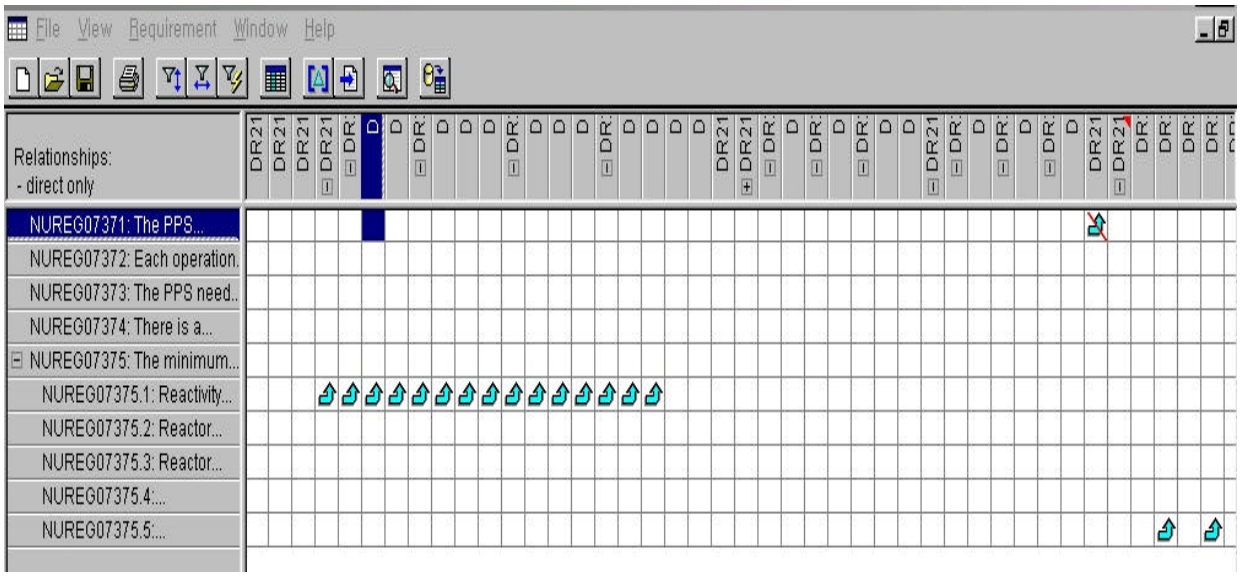

Figure 4. The example of the RTM using Requisite Pro

3.4 Code Verification

The code produced for the ADRPS is inspected to ensure that it correctly implements the design and has not introduced any errors. Techniques to be applied to code verification for the ADRPS will be 'eye-balling' (or desk checking) and walk-through[3].

3.5 Testing of the ADRPS

Software tests for the ADRPS will be performed on modules, on group of modules and on the entire software specified in the SRS for the ADRPS[4]. The software tests must be performed according to a written test plan. The test plan may be part of the V&V plan or may be referenced in the V&V plan. The output documentation of this activity is the software test report.

The QNX OS and the VxWorks are selected for software diversity for the ADRPS. Testing of the ADRPS can cost a lot of effort if the number of tests is high enough that meaningful reliability figures can be derived. The search for system structure that requires a smaller number of test runs leads to diverse arrangements. The ADRPS is a diverse system which contains several independently developed real time software packages that solve the same problem. Their results are compared with one another before any output is made. If the comparison of results leads to no discrepancy among the independent computation outputs, one of the output area is transmitted. If discrepancies are recognized, a warning is issued.

3.5.1 Unit Testing

The basic modules for the ADRPS are tested separately first. Basic coding errors should be detected by this method. For testing purposes, a test environment will be provided. It consists of a test driver, which provides the module with the required input parameter values plus additional necessary interfaces. Testing techniques like path testing will be applied during the unit testing. Features that will be tested include each data-flow path, and testing on around each boundary. The unit testing for the ADRPS will test the executable code integrated with target hardware and pre-developed software meets all of the requirements specified in the SRS and behaves as specified in the SDD.

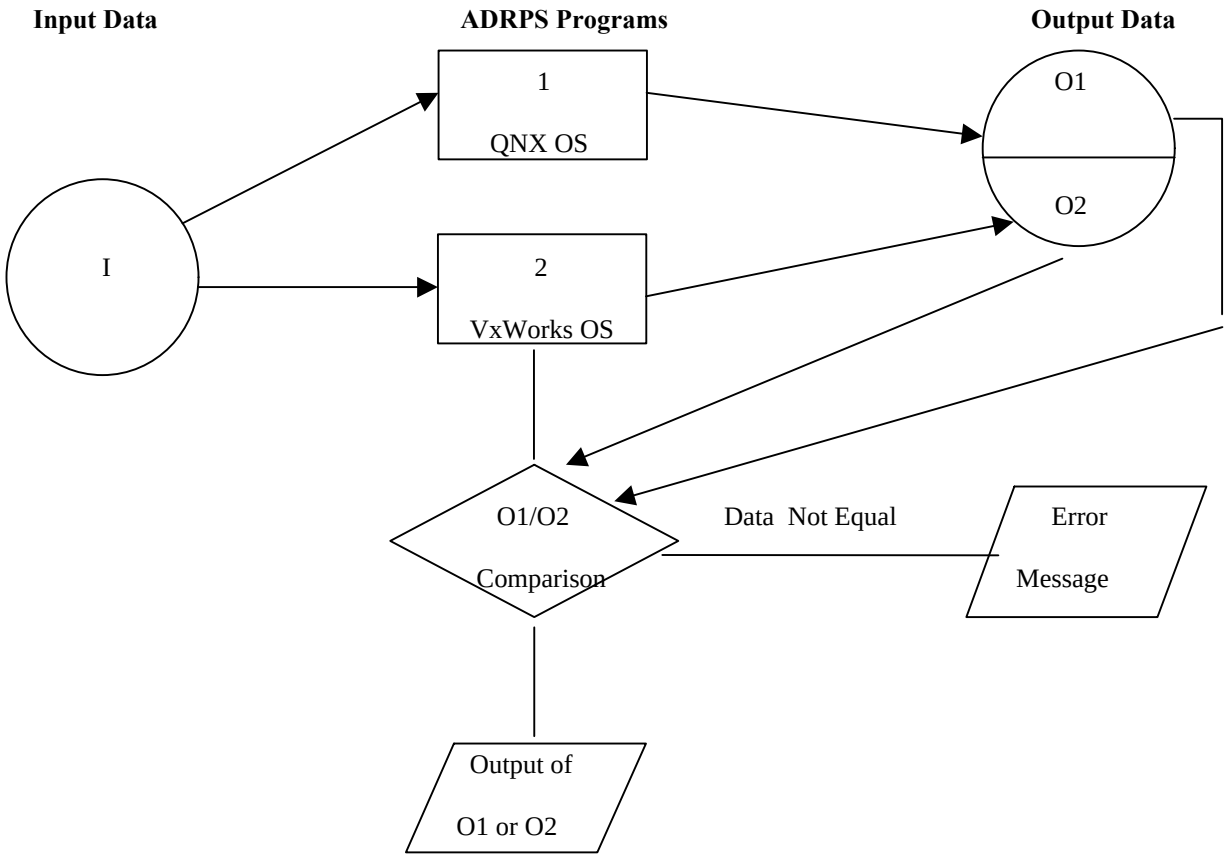

Figure 6. Test Structure of the ADRPS

3.5.2 Integration Testing

After all the basic modules for the ADRPS have been tested separately, integration testing starts. Some of the tested modules which communicate with each other are connected to a subsystem and tested. This is done for all subsystems. Again a test environment for integration testing for the ADRPS will be provided. The test case selection has to make sure that all the newly connected interfaces are tested sufficiently. During unit testing the real-time aspect most often is to be neglected, it has to be taken into account during integration testing.

3.5.3 Validation Testing

Validation testing for the ADRPS ensures that the ADRPS software, when integrated with the ADRPS hardware, meets the requirements specified in the FRS. Static and dynamic simulations of input signals can be arranged to operate the ADRPS computer system in modes which represent normal operation, anticipated operational occurrences and accident conditions which require the ADRPS action. Each function for the ADRPS should be confirmed by conducting a representative test for the actions caused by each parameter for trip or protection,

singly and in combination. The validation test design should therefore define the required input signals with their sequences and values. The validation test cases must describe the anticipated output signals with their sequences and values and the signal level acceptance criteria.

4. Software Verification and Validation Structure for the ADRPS

4.1 Organization

Engineering & Development Organization for the ADRPS contains system design team, software design team and software verification team. The manager of the ADRPS manager is responsible for the overall development and integration of the ADRPS and associated software. All system and software development workers report to the ADRPS manager. System Design Team (ST) develops functional requirements, hardware requirements and performance requirements for the ADRPS. Software Design Team (DT) develops software design requirements and software for the ADRPS based on requirements from system design team. Software Verification Team (VT) performs verification and validation activities for the ADRPS software. In order to improve software quality for the ADRPS, software V&V is performed in a manner which maintains V&V process independent of the design process. Confirmation of compliance with the independence requirements can be accomplished by VT performing V&V activities independently and by separating VT from design group organizationally. Communications between VT and design group shall be documented in written reports.

4.2 Resource Responsibilities

The suggested Software Verification Team consists of following elements.

4.2.1 Lead Verifier

The lead verifier is the team leader responsible for all technical matters concerning the verification of the ADRPS. The lead verifier has the responsibility for the development of the verification requirements and test plan. It is also the responsibility of the lead verifier to check the documentation compiled by the design team for completeness and unambiguity.

4.2.2 Verifier

Verifiers check the portions assigned to them with the use of the verification requirements and test plan documents. These checks are carried out by the verifier with any of the appropriate tools and techniques discussed that are agreed upon with the lead verifier.

4.3 V&V Tools, Techniques and Procedures

All tools, techniques and procedures used for V&V functions shall be documented. This is particularly important because the efficiency of the entire process is greatly increased through the use of specially developed custom computer tools to process program source code and documentation[2].

4.4 Software Life Cycle V&V

4.4.1 Concept Phase V&V

Concept phase V&V is the period prior to formal definition of the ADRPS requirements, which may include a feasibility phase. No formal V&V requirements for the ADRPS exist within this phase until a development task requiring V&V is required. Preparation of a V&V plan for the ADRPS, including schedule and personnel requirements should be developed at this time.

4.4.2 Requirements Phase V&V

The requirements phase of the ADRPS is the stage in which the requirements of the software (functionality, performance, design constraints, attributes, testability, external interfaces, etc.) are specified, documented and reviewed. Tasks completed during the requirements phase include software requirements specifications, and interface specifications.

4.4.3 Design Phase V&V

The design phase of the ADRPS is the period in which the software is designed, documented, and reviewed in accordance with previously established requirements. Tasks completed during the design phase include software unit test plans, integration test plans, and a software design review to ensure that requirements are satisfied.

4.4.4 Implementation Phase V&V

The implementation phase of the ADRPS is the period in which the design is translated into programming language, and the implemented software is reviewed and tested to identify and correct coding errors. Tasks completed during the implementation phase include software coding listings, software unit verification test and anomaly reports, configuration management implementation report, and hardware/software integration test plan.

4.4.5 Testing Phase V&V

The test phase of the ADRPS is the stage during which the hardware and software are integrated together as a system and tested. Integration testing is performed in accordance with test procedures, and results are analyzed to ensure that the hardware and software components function correctly together and that the system fully satisfies the requirements. Tasks completed during the test phase include software build documents, integration test report, and validation test plan.

5. Conclusions

Software V&V methodology to be applied to the ADRPS was described in this paper. Software V&V for the ADRPS is the formal act of reviewing, testing or checking, and documenting whether the ADRPS software components comply with the specified requirements for a particular stage of the ADRPS development phase.

To assure quality of the ADRPS software, the software should be developed in accordance with software development procedures and rigorous software V&V should be

performed. One of the ADRPS goals is to prevent the CMF. However, there is a possibility that the designers of the ADRPS will make errors and that the errors will result in undetected faults in the ADRPS.

The ADRPS may then be installed with faults in it and will result in the CMF. The failures that can result from these faults make it necessary to apply active measures in a systematic manner to find and correct errors. Therefore, rigorous Software V&V is also required to develop the ADRPS software. Outputs from the ADRPS will be demonstrated that the use of this software V&V methodology results in a high quality, cost-effective product.

References

1. W.J.Quirk, 1985. Verification and Validation of Real-Time Software

2. Steven R. Rakitin, 1997. Software Verification and Validation

3. IAEA, 1999. Verification and Validation of Software Related to Nuclear Power Plant Instrumentation and Control

4. Glenford J. Myers, 1979, The Art of Software Testing

5. USNRC, NUREG/CR-6101, 1993. Software Reliability and Safety in Nuclear Reactor Protection Systems

6. EPRI, TR-103699-V1, 1994. Programmable Logic Controller Qualification Guidelines for Nuclear Applications

7. KOPEC, 00000-IC-SM717, Rev.00, 2000. Software Development Manual for I&C Systems

8. KOPEC, Sang Gu Nam, 2000. Functional Requirements for the Advanced DRPS for National Research Lab.