DISTRIBUTED CONTROL SYSTEM FOR CANDU 9 NUCLEAR POWER PLANT

HARBER J.E., M.K. KATTAN, M.J. MACBETH

AECL
446A 2nd Avenue North,
Saskatoon, SK., S7K 2C3.

Abstract

Canadian designed CANDU pressurized heavy water nuclear reactors have been world leaders in electrical power generation. The CANDU 9 project is AECL's next reactor design.

The CANDU 9 plant monitoring, annunciation, and control functions are implemented in two evolutionary systems; the distributed control system (DCS) and the plant display system (PDS). The DCS implements most of the plant control functions in a single hardware platform. The DCS communicates with the PDS to provide the main operator interface and annunciation capabilities of the previous control computer designs along with human interface enhancements required in a modern control system.

Introduction

In previous CANDUs, plant control was performed by control computers, analog devices and relay logic. System control was performed by dual redundant computers which executed a set of control programs for monitoring, annunciation, and control of plant systems. In a second level, control devices such as analog controllers and programmable logic controllers (PLCs) handled lower level control functions. The application programs for the control computers were written in low-level programming languages such as assembler. The lower level device control logic was written in a functional block language or was performed in hardwired logic.

The requirements for the DCS are based on the proven strategies of previous CANDU control system designs. The design architecture of the DCS (see Figure 1) is comprised of a set of control and monitoring stations, known as partitions, interconnected to each other (and the PDS) by communication links.

DCS application programs are implemented using a function block language (FBL) for all hierarchial levels in the control system. The use of a FBL and the application of software engineering quality assurance procedures leads to an efficient and more streamlined software development process.

Plant control functions are assigned to the individual partitions, based on an independence assessment of the control programs. As part of the CANDU 9 design process, a functional analysis considering safety, reliability, and maintainability considerations has been prepared to outline the basis for the DCS hardware platform. In addition, an independent hazards analysis of the DCS application will be conducted to confirm the extent of failure impact.

Each partition will be developed by selecting off-the-shelf modules from the manufacturer's existing product line. The basic configuration of all partitions are similar (see Figure 2), but have some differences based on the input and output signal requirements and the processing requirements of the application programs. DCS application programs are implemented using a function block language (FBL) for all hierarchical levels in the control system. The use of a FBL and the application of software engineering quality assurance procedures leads to an efficient and more streamlined software development process.

The CANDU 9 design process applies one allocation philosophy for plant functions to a hardwired control system (referred to as a hard system) or a computer based software control system (referred to as a soft system). This philosophy is based primarily upon station safety requirements with a secondary evaluation of potential economic benefits provided that the safety constraints are satisfied. If a control/indicating/annunciation function is required to manoeuvre the station from an event end point following a potential failure of a soft system, then that function must be provided by a hard system.

The CANDU 9 hard systems will be unequivocal such that functions designated as requiring a hardwired configuration will be implemented using 100% hardwired, discrete devices. For example, the plant system functions required to safely take the unit from a soft system failure end point (e.g. zero power hot state) to the cold shutdown state will be hardwired with no data transmissions via computer devices.

A hard/soft allocation review will be conducted for those system functions which are presently provided on the reference plant but which are not restricted by safety constraints. A carefully developed hardwired versus soft based control rationale ensures that operator awareness of system extent and functional intent is obvious and interfaceable at all times.

Function allocation is considered early in the requirements definition stage as designers are guided to consider, for example, if the function should be performed automatically or manually (i.e. allocated to machine or human) and if automatic, should that function be performed by computer or hardwired devices. The procedures, design guides and reference plant basis assessment documents aid the designers in this allocation process. Further function allocation details are defined as the system design description is prepared.

As the design proceeds, this control information can be revisited and completed to a greater level of detail so that such details as automatic/manual, location in main control room/secondary control area, process/safety/post-accident monitoring/critical safety parameters, CRT-based or hardwired, plant operating regions, and so forth can be addressed. Hardwired conventional control and monitoring components will be independent of PDS and DCS are thus available on the main control room panels to allow plant shutdown and cooldown operations in the event of a failure of PDS or DCS. All of the necessary controls needed to manoeuvre the plant from a hot pressurized to a cold depressurized state are provided by hardwired devices on the main control room panels.

CANDU 9 DCS Architecture

The DCS is designed using ABB Procontrol P13/42 products. ABB P13/42 distributed control systems use two types of bus for communication, a local bus and an intra-plant bus. The local bus is a card cage backplane which may be extended over multiple card frames up to 30 metres by extension cables and amplifiers. Collectively, all equipment on a single local bus is referred to as a P13 station.

The intra-plant bus, or data highway, utilises coaxial cable, and provides a communications backbone to integrate P13 stations into a complete system, and to communicate with foreign systems. It can be up to 1,400 metres in length. Data on both busses is exchanged in the form of telegrams, each of which contains an address and a single sixteen bit data word. The data word may represent a single analog signal, or 16 binary signals.

Modules (field replaceable units) which pertain to the local bus are referred to as P13 equipment, while modules which pertain to the data highway are referred to as P42 equipment.

P13/42 modules fall into five broad classes:

a) Bus controllers, which generate sequences of addresses and other bus control signals on each bus.

b) Process input modules, such as analog and digital input modules. Each process input module has one or more local bus addresses assigned by front panel switches. Process input modules listen for their address, then write their data to the local bus.

c) Process output modules, such as analog and digital output modules. Each process output module has one or more local bus addresses assigned by front panel switches. Process output modules listen for their address, read the data which follows, and update their outputs accordingly.

d) Processors receive and send data via the local bus. Each processor has a unique identification number assigned by front panel switches which is used to identify the processor for maintenance purposes. The local bus interface section of the processor listens for the source address of data required by its application program, then reads the data from the local bus into a data buffer. Control computations are based on the data in this buffer, and the results of the computations are written to the buffer. The local bus interface listens for the address of sinks to which it needs to send its output data, then writes the data to the local bus.

   The processor design permits two of them to be configured for dual redundant operation in a manner which is transparent to the application software. The active processor in a redundant pair continuously updates the standby processor with past state values, so that switchover, when required, will be bumpless.

   If necessary, the application program can be written to take specific action when it assumes control. For example, the present process values could be used to determine the setpoint, pending confirmation by the operator of a manoeuvre which might have been in progress at the time of the failure.

e) Bus couplers transfer data to and from the local bus. Some couplers interface the local bus with the Data Highway, while others interface the local bus and third party systems via a serial communications link.

f) PC interface cards are used to interface the data highway directly to foreign systems via a PC station. The card also performs time stamping on all data leaving the DCS data highway.

In addition, there are the busses themselves, and the other hardware modules associated with them, such as access couplers and terminators.

System Level Architecture

The system level architecture is shown in Figure 2. The DCS is divided, tentatively, into five functional partitions. Each partition will contain dual redundant data highways DH_{Xn} and DH_{Yn}, where 'n' is the partition number. These highways provide a communications backbone which carries all traffic within partitions, and also carries data to and from the PDS and other partitions.

Communication with the PDS is implemented using PC interface cards. Each partition will have a pair of PC interface cards, one of which will be located in gateway 'X' and one in gateway 'Y'. The DCS/PDS gateways are each comprised of two general purpose workstations incorporating an industry standard architecture (ISA) bus. The PC interface cards will provide time stamping of the change of state for binary signals. Each partition will receive identical operator commands from the PDS, and place them on the appropriate partition data highways DH_{Xn} or DH_{Yn}. The control status resulting from these commands (e.g. control modes, "soft" handswitch positions, setpoints) will be retained in the NOVRAM of the appropriate processors, so that it will be available following a processor restart.

Inter-partition communication will be via transfer stations, one of which will handle the X data highways and one the Y data highways. These transfer stations carry setpoints and other data required for integrated plant control from one partition to another. The transfer stations are local busses, each populated with one bus coupler for each partition requiring inter-partition communication, and one local bus controller. Both transfer stations would have to fail before inter-partition communication is lost.

The partition level architecture is shown in Figure 4. Two redundant data highways, DH_{Xn} and DH_{Yn}, provide communication between group and device levels, and among channels at the device level. Synchronization between the two data highways is not required. It is possible, but not desirable, to run the data highways in a synchronized configuration, as this would increase the probability of a common mode fault affecting both highways simultaneously.

Group level control functions are implemented in the non-channelized group level station. All inputs and outputs at the group level are expected to be routed via the partition data highways.

Process system inputs are scanned periodically in the channelized device level stations DLS_{An}, DLS_{Bn} and DLS_{Cn}. Device level controls and process system outputs are implemented in DLS_{An} and DLS_{Cn}. The device level equipment for channels A, B and C are located in separate cabinets. The bus couplers located in the device level stations will map the X and Y data streams onto separate local bus addresses. Selection of redundant signals will be performed by the control processors. DH_X data will be used if it is available, otherwise DH_Y data will be used.

Power for X equipment and for channel A device level stations will be from electrical power supplies for channel A. Power for Y equipment, and for channel C device level stations will be from electrical power supplies for channel C, similarly. Power for channel B equipment will be from electrical power supplies for channel B. Dual power supplies with combined outputs are used throughout.

All channels perform data acquisition for their associated process sensors. This data is passed to the control processors, either directly on the local bus, or via the data highways. Redundant sensors for critical input signals will be channelized.

The effects of failures of a single module causing failures to multiple process system loops will be confined to a single process system with suitable annunciations. The design process will ensure that only inputs pertaining to related subsystems are assigned to a single input module. Similar precautions will be observed for output modules.

Channel B is used for acquisition of the channel B inputs for process signals, and for scanning of fast digital inputs required for sequence of events reporting. Channel B has no process output or processing capability.

Process outputs are implemented in the channel A and C control stations. Wherever a fail safe condition of the process system is clearly identifiable, this condition will be selected to be the de-energised state of the output module.

Group and Device Level Control Functions

Group control functions will have one or more of the following characteristics. They may:

a) require input from several sources;

b) implement relatively complex logic;

c) drive several devices;

d) contain control logic which changes significantly depending on the mode or operational state of the controlled system;

e) have outputs which drive non-redundant actuators (e.g. liquid zone control level valves, which must virtually all function to avoid functional failure); or

f) have outputs which drive actuators which are redundant from a safety point of view, but not from a production point of view (e.g. parallel, fail open feed valves).

Device control functions will have one or more of the following characteristics. They may:

a) have control loops which are relatively simple, and involve a small number of inputs;

b) act on setpoints or error signals computed at the group level;

c) provide a facility for direct operator override in an output loop; or

d) monitor redundant analog outputs or devices.

System Reliability

Loss of a partition is considered to have occurred when any of the following capabilities have been lost:

a) processing at the group level,

b) both partition data highways,

c) communication with PDS,

d) loss of certain critical inputs or outputs.

The MTBF for loss of a partition is 216 years. This MTBF reduces to 170 years if two device level stations are necessary in channels A and C due to local bus address constraints, but the number of I/O modules is held the same. Assuming that loss of any one of the five control partitions leads to a plant outage, the estimate for this event is one fifth of the figure calculated for loss of one partition or 34 years.

CANDU 9 Control/Display/Annunciation Strategy

A major evolutionary change from previous CANDUs is the separation of the control and display/annunciation features formerly provided by the digital control computers (DCCs). This CANDU 9 function separation provides control in the DCS and display/annunciation in the PDS. This strategy allows powerful computers without application memory constraints or execution limits to provide extensive display and annunciation enhancements within an open architecture. The DCS implements most of the plant control functions on a single hardware platform. The DCS communicates with the PDS to provide the main operator interface and annunciation capabilities of the previous control computer designs along with human interface enhancements required in a modern control system.

The PDS emulates the display, annunciation, data logging and supervisory functions provided by the DCCs of previous designs, but with enhanced capacity and performance. The system will maintain a plan-wide real time database containing all plant variables, annunciation control flags, engineering units conversion equations, etc.

Monitoring/calculation functions include flux mapping and channel temperature monitoring. These control-related computations are allocated to the PDS to unload the control platform from such background mathematical manipulations. The time constants involved are such that the additional communication delays are acceptable.

Mock-up and Developmental Test Facility

A mock-up of the control centre panels and consoles in the AECL design facilities will be used for verification and validation of the CANDU 9 design features, controls, displays and operator interactions. The functionality of the simulation supported CC mockup provides a mechanism for on-going verification and validation (V&V) design activities by system designers throughout the entire project design life-cycle. The verification process includes traditional supervisory and peer document reviews, CADDS reviews, procedural walk-throughs moving to validation by utilizing the physical full scale panel mock-up facility which is supported by the PC based CANDU 9 plant simulation. The CC mockup serves as a designer tool to verify that the individual system designs conform to human factors engineering (HFE) principles, ensuring acceptable performance of specified operational tasks.

During the project design integration phase, significant portions of DCS partitions will be prototyped in the CANDU 9 control centre mockup facility. The DCS development system is used in support of the DCS architecture design. Two general groups of tests have been identified;

a) tests of specific DCS design features to confirm the architecture will satisfy specific CANDU 9 DCS functional, performance, failure effects and detection, and PDS/DCS interface requirements; and

b) a dynamic demonstration of the CANDU 9 DCS control programs. This program test is intended to illustrate project software development procedures and serve as further support of the architecture concept.

The tests have been selected to demonstrate features which may pose some degree of risk or uncertainty. These tests exclude any requirements which will be verified by analysis, or which can be verified by reference to the supplier's user manuals or other information sources.

The DCS development system is interfaced to the CANDU 9 PC based plant simulator (Reference 1) through input/output modules. The simulator is used as a tool in the DCS design and testing process. In some tests, the equivalent control algorithm in the simulator will be disabled, and the process model input/output signals will be routed to the DCS prototype. In this way, the DCS can control the plant model and exercise the control application program to verify the functionality of the control program.

The DCS development system is also interfaced to the CANDU 9 PDS (including the computerized annunciation system), the plant wide common database, the mock-up panel and console hardwired devices. Figure 5 provides an overview of the interfaces to the DCS development system.

This combination of DCS development system and simulation is used for control strategy implementation, analysis of overall plant control performance, estimates of tuning parameters for major control loops, and evaluating the interaction of the DCS control programs with the plant display system. This means that a DCS control program application feature can be evaluated, tested and confirmed by dynamic simulation exercising initiated by operator actions such as soft based setpoint entries or hard based actions such as handswitch positions.

A suite of tests including plant state, failures, loss of support systems and initiating events can be applied systematically to confirm the DCS architecture, communications, application code and control strategy within the broader context of operator information and information presentation needs to ensure optimum performance and operability.

Preliminary tuning limits and default values can be established in a practical manner during the design phase to facilitate work and work processes during the commissioning stages. Integrating the operational and maintenance development systems allows a very detailed co-ordination strategy to be evolved to ensure that the correct information is presented in the correct format with the needed level of detail for the intended user.

DCS Software Design

An overview of the software development and verification process is given in Figure 5. The figure shows how the detailed design and code evolves from the monitoring and control requirements of the process system, how each stage of the design process is subject to review, and how the design input documentation (DID) is used as the basis for subsequent verification and validation activities.

One of the primary determinants of the quality requirements for development of the DCS software is the safety criticality level of the process application. A DCS software criticality assessment will be performed towards the end of the design integration phase, when necessary information on the safety role and impact of failure of the systems being controlled by the DCS is available. The output of this assessment will be the DCS software categorization report.

Project Status

At the time of writing this paper, the DCS design requirements and software development process have been approved and issued for use. Concepts for the DCS hardware design are being developed and tested on the prototype system in the AECL control centre mockup.

Nuclear regulatory information sessions have been completed and the regulatory review is nearing completion. A representative design slice of DCS application software is being developed to demonstrate the process of software development and review using a function block language for process control.

References

(1). KATTAN M.K., M.J. MACBETH AND K. LAM, "CANDU 9 Nuclear Power Plant Simulator," 19th CNS Simulation Symposium, Oct 1995, Hamilton, Ontario.

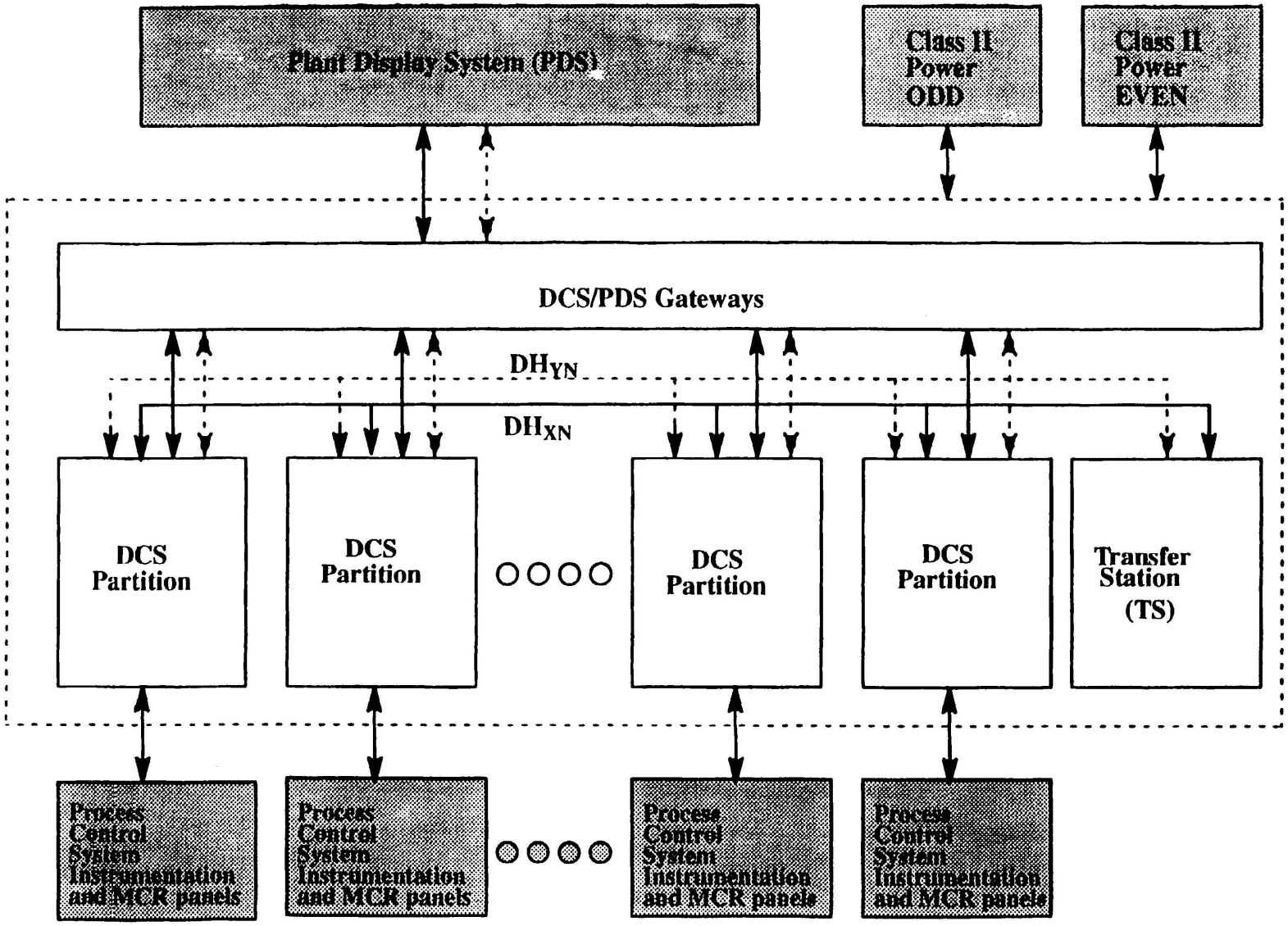


Figure 1 – Overall DCS Configuration

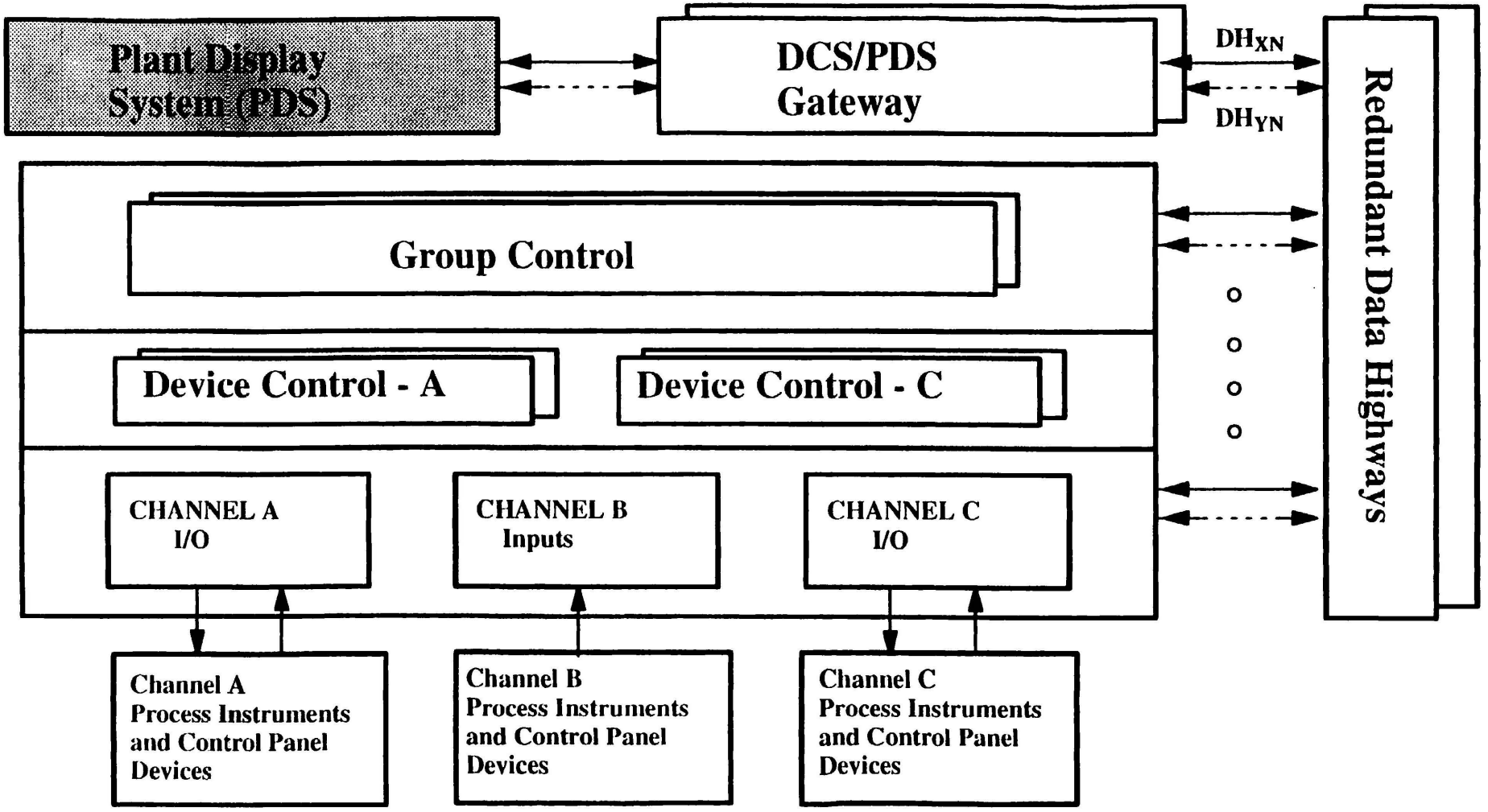


Figure 2 – DCS Partition Concept

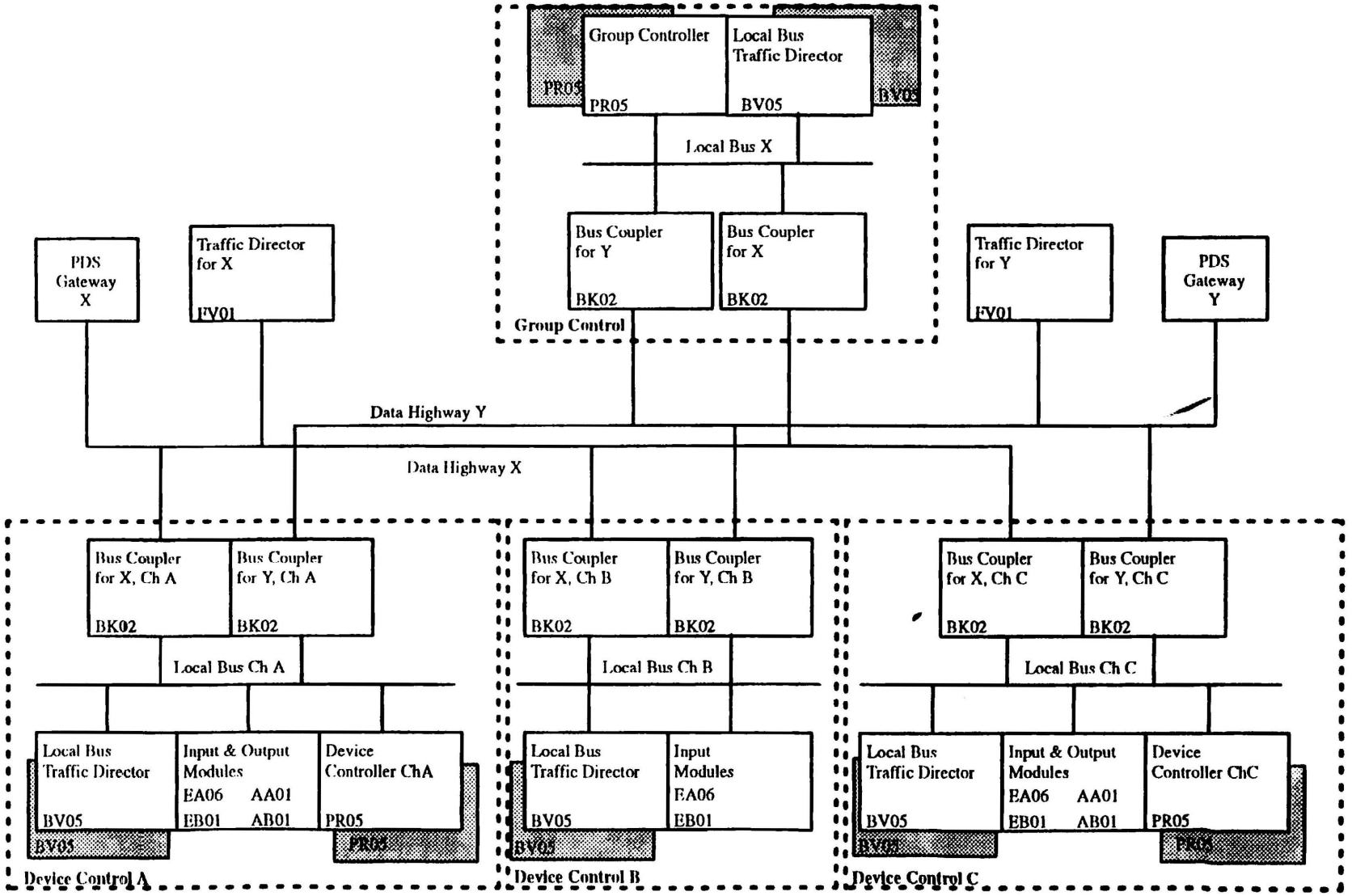
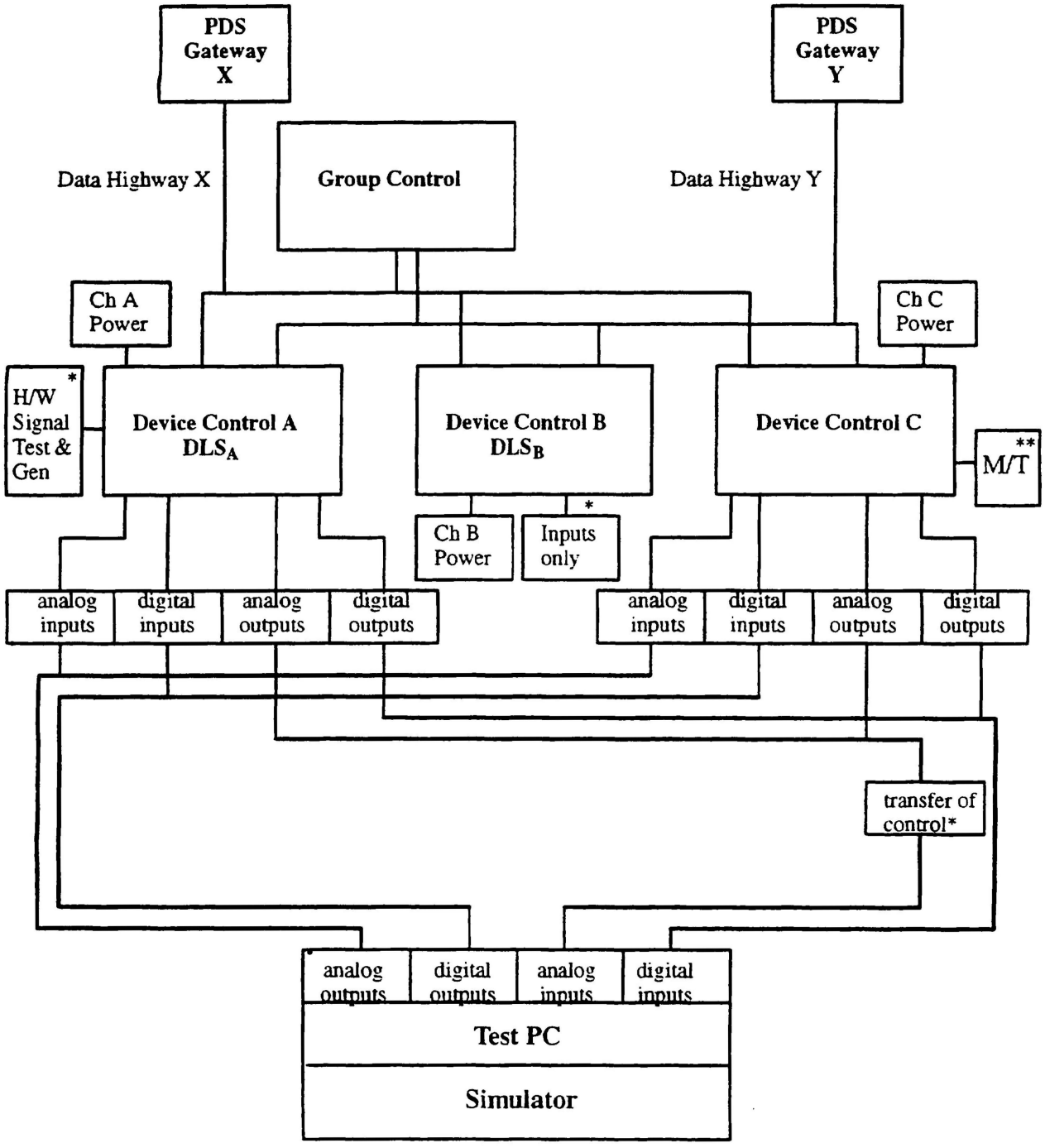


Figure 3: DCS Partition Architecture



* As required

** M/T is a monitoring and test program, capable of reading and writing to DCS software signal registers

Figure 4 – DCS Development System Interfaces

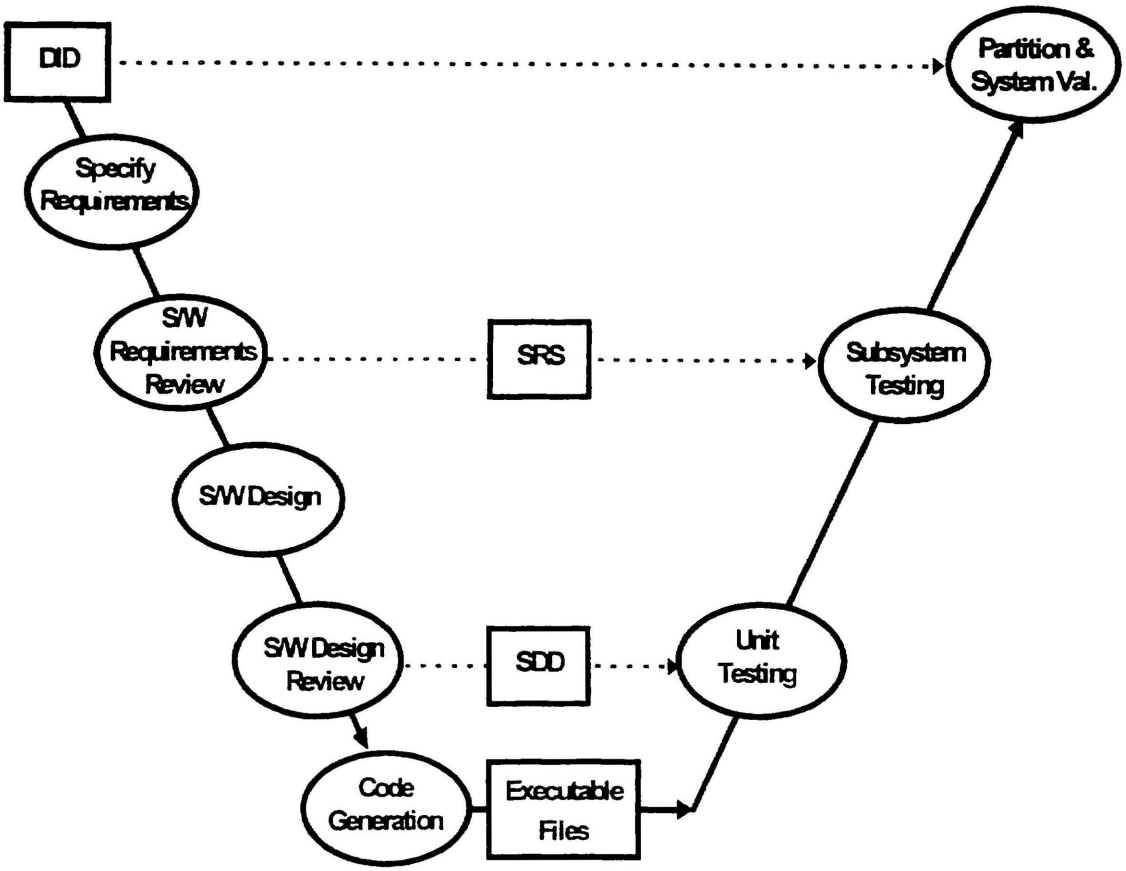


Figure 5: Overview of the Software Development and Verification Process