THE CANDU ® 9 DISTRIBUTED CONTROL SYSTEM DESIGN PROCESS

J.E. HARBER, M.K. KATTAN
Atomic Energy of Canada Limited
2251 Speakman Drive, Mississauga, Ont., L5K 1B2
and
M.J. MACBETH
Institute for Advanced Engineering/Ajou University
Energy Systems Research Center
Suwon, Republic of Korea, 441-749

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.

Plant control for the CANDU 9 station design is performed by a distributed control system (DCS) as compared to centralized control computers, analog control devices and relay logic used in previous CANDU designs. The selection of a DCS as the platform to perform the process control functions and most of the data acquisition of the plant, is consistent with the evolutionary nature of the CANDU technology. The control strategies for the DCS control programs are based on previous CANDU designs but are implemented on a new hardware platform taking advantage of advances in computer technology.

This paper describes the design process for developing the CANDU 9 DCS. Various design activities, prototyping and analyses have been undertaken in order to ensure a safe, functional, and cost-effective design.

Introduction

The next generation CANDU power station is being designed as a single 900 MWe unit. This evolutionary design includes enhanced versions of existing CANDU control functions and advanced new features made possible by new technology.

The CANDU 9 plant monitoring, annunciation, and control functions are implemented in two evolutionary systems; a plant display system (PDS) and a distributed control system (DCS). The DCS implements most of the plant control functions in a common hardware platform. The DCS communicates with the PDS, which provides the main operator interface and annunciation capabilities of the previous control computer designs along with human interfacing enhancements required by a modern control system. A context diagram of the DCS is shown in Figure 1. The design of the CANDU 9 DCS is described in Reference 1.

In existing CANDUs, plant control is performed by digital control computers (DCCs), analog control devices and relay logic. At the highest level, system control is performed by dual redundant computers which execute a set of programs for monitoring, operator display, annunciation, and control of important plant systems. At a lower level, control devices such as analog controllers and programmable logic controllers (PLCs) handle individual device control functions. The application programs for the control computers are written in low-level programming languages such as assembler, while the lower level device control logic is written in a symbolic language or is performed in hardwired logic.

While the DCS technique has been applied in a number of limited scope applications in CANDU nuclear power plants, this is the first application of a full-scope plant DCS that provides a replacement for most of the conventional control equipment and DCCs used on existing plants. Special attention has been paid to all areas of the CANDU 9 DCS design process to ensure that the system hardware and software design requirements are specified completely and adequately. This process will eliminate implementation risks, and prevent operability or maintainability difficulties while ensuring a quality product.

® CANDU is a registered trademark of Atomic Energy of Canada Ltd.

This paper describes:

- the distributed control system design cycle, outlining the various steps involved realizing the design of such a system,
- analysis methods used to support the hardware and software design, the distribution of the application software into the various control levels within the DCS, and the hazard and operability analysis,
- the guidance provided to the control application system designers to allocate process control system applications into software based (DCS) applications and hardwired applications,
- the design, verification and validation processes for the application software, and
- the qualification activities for pre-developed software.

The Distributed Control System Design Cycle

The engineering effort for a DCS design is managed in a similar fashion to other instrumentation and control projects involving both hardware and software. Typically, such projects involve the following activities:

1. outlining the DCS project goals and objectives,
2. preparing the DCS project schedule, deliverables, and milestones,
3. listing the design requirements,
4. preparing a description of the design,
5. developing the hardware technical specifications and procurement documents,
6. preparing system process and instrumentation diagrams, instrument loop diagrams, elementary diagrams, and cabling and wiring diagrams,
7. where software is involved, preparing of software functional specifications and design, verification and validation of the software, and
8. preparing commissioning, operating and maintenance manuals.

In the case of the CANDU 9 DCS design, a number of analyses have been identified, in addition to the above list of activities, which are either specific to a DCS, or provide more additional information than previously required due to the application of a DCS. Some preliminary analyses had to be performed early in the design cycle in order to assist with the conceptual design of the DCS. These early analyses included preliminary assessments of the DCS architecture structure, the software reliability, and the adequacy of independence and separation provisions for monitoring and control functions of plant systems using the DCS.

In general, design documents, such as the design requirements and design description, evolve and undergo revisions as the design progresses. These revisions will incorporate the approved findings of final system analyses and feedback from the standard design verification activities such as formal peer review and design review meetings.

Nuclear regulatory information sessions have been held and a regulatory review was completed indicating that there were no barriers to licensing for the CANDU 9 DCS. A representative design slice of DCS application software for the Liquid Zone Control system has been developed to demonstrate the process of software development and review using a function block language for process control. Prototype testing, using a functional DCS partition, has been performed to verify the functionality and performance of the various options considered in hardware and software design.

DCS Design Analyses

Software Categorization

The safety category of all software in a nuclear plant must be established by examining the safety significance of the plant systems involved and the possible software failure impact on these systems. Four categories are defined in Reference 2, with category 1 having the most serious consequences of failure and therefore the most stringent QA requirements, and category 4 having no safety significance at all. The software category and resulting software engineering development process are important factors in determining the overall cost and schedule of the DCS software.

A preliminary DCS Software Categorization Assessment was performed early in the conceptual design stage, based on preliminary design information about the safety role and impact of failure of the systems being controlled by the DCS. The finding of this assessment resulted in the categorization of some DCS software as Category 2, with the remainder as Category 3. Following the completion of the probabilistic safety assessment, and prior to the start of the software requirements specification and design, a final software categorization assessment will be performed.

Functional Independence Assessment

The DCS architecture provides a number of independent segments, known as partitions. The many plant monitoring and control functions can be allocated to different partitions to provide more separation and independence. Using a preliminary probabilistic safety assessment for the plant systems as a starting point, a preliminary functional independence assessment was performed to determine which plant systems monitored and controlled by the DCS are required to be independent of one another.

Based on these independence requirements, control and monitoring functions are assigned to the various DCS partitions. These assignments will be confirmed by a functional independence assessment when the probabilistic safety assessment has been completed.

Architecture Analysis

Preliminary failure and response time analyses have been performed in support of the design of the resulting DCS partition. These analyses showed the compliance of the DCS design with the plant reliability and application software timing requirements.

Hazard Analysis

The hazard analysis assesses the impact on plant safety and production of DCS software and hardware related failures, operator actions, maintenance activities, software allocations between partitions, and the hierarchical control levels within the DCS. During the DCS design process, various hazard analyses are performed:

1. DCS platform hazard analysis: This assessment addresses the hazards and operability issues associated with the DCS architecture and of the physical modules used to configure it. A preliminary analysis has already been carried out to identify issues associated with the use of a DCS for monitoring and control functions using information from the preliminary DCS conceptual design.
2. Process control system hazard analysis: This assessment is conducted for each plant safety related system which makes use of the DCS for monitoring and control purposes. This analysis is carried out following the preparation of the software functional requirements for applications controlled and monitored by the DCS.
3. DCS partition hazard analysis: This assessment is carried out for each finalized DCS partition. The analysis concentrates on common-mode failures which could affect more than one plant system within that partition. This assessment is carried out following allocation of software to processors and inputs and outputs to specific input and output modules.
4. In parallel with the DCS overall integration, testing and validation activities, a total system hazard review will be carried out to address very high level inter-partition issues.

The outcomes of each of these assessments are fed back to the input design documentation for revision as appropriate and a follow-up analysis with limited scope is performed before moving on to the next step in the design cycle.

Guidance to Control Application System Designers

A control application design guide will provide assistance to system designers in areas such as defining the scope of the DCS for each control application, the need for hardwired backup controls on safety-related end devices, the definition of signals required to be displayed at the operator interface, and the specification of various features in the DCS design such as redundant inputs and outputs for control signals.

For example, the application design guide will assist the system designer in selecting the appropriate method of control to be provided to the plant operator. The operator can interface with the plant control system in three ways:

- interfacing to computer control provided by the DCS/PDS,
- interfacing to analog supervisory control stations, and
- manually controlling the end device through panel controls.

The selection of the appropriate interface will be based on rules and examples provided in the application design guide.

Design, Verification, and Validation Processes of the Application Software

Software Functional Requirements

The control and monitoring requirements for the DCS, based on previous CANDU designs, will be defined by the system designers for the individual process systems. This definition is prepared in a procedurally bounded format with particular attention to completeness, accuracy and unambiguity of the requirements. The requirements are presented in either textual, graphical, or truth table format which best defines the information.

Two types of documents are used for defining software functionality:

- Program Functional Specifications (PFS) are used to describe complex computations; and
- Device Control Requirements (DCR) are used for coordination of devices using simple logic that consists of independent loops with limited numbers of inputs and outputs.

The Design Input Documentation (DID) for each partition is developed by collecting all of the functional requirements for the systems allocated to the partition, taking the results of the functional independence assessment analysis into account.

The definition of the plant control systems functional requirements is the first step in the design of the application software. The actual design of the software in the DCS is governed by the quality project plan (QPP) for the DCS design.

Quality Project Plan (QPP)

The DCS QPP specifies the systematic process to be used for the production of the DCS software from the functional requirements. The scope of work accomplished under the QPP is the development and verification of all software within the boundary of the DCS until the plant is declared to be in-service. The QPP does not define an engineering process for DCS hardware design, but describes the activities and responsibilities of the hardware design team where they relate to software engineering.

The DCS QPP calls for the preparation of procedures to guide the software design and verification processes. These procedures achieve consistency in the design methodology by various designers, ensure adherence to certain design and verification principles, and provide correct and complete documentation.

The QPP also defines how the independence amongst the design and verification activities is achieved. Personnel assigned to design and verification activities would be drawn from a common staff pool, but verification activities associated with a particular software element will not be assigned to any person who was directly involved in the design of the element. Validation of functional requirements would be performed by reactor and process control functional designers. Validation of requirements derived from the DCS hardware design would be performed by DCS hardware designers.

An overview of the software development and verification process is provided by Figure 2. This figure shows how the detailed design and resultant code for each partition evolves from the design input documentation, how each

stage of the design process is subject to review, and how the design documentation is used as the basis for subsequent verification and validation activities.

Software Requirements Definition

Requirements for the DCS software are derived from the Design Input Documentation (DID). The Software Requirements Specification (SRS) presents a formal, unambiguous statement of the DID requirements as they apply to the DCS software. Interface requirements, for example, are expressed in terms of the electrical signals which are present at the DCS input/output terminals. There will be one SRS for each DCS partition.

The requirements contained in the SRS will be implemented and documented as Function Block Diagrams (FBDs). Overview diagrams provide an hierarchical view of the software. Detailed diagrams describe the requirements for each partition, either using graphical primitives (AND gates, integrators, etc.), or using macro blocks, whose functionality is in turn described using graphical primitives or some other formal method. This representation has the considerable advantage of being easily reviewable by system designers who are not programming experts.

Approved requirements are entered into the Requirements Tracking and Verification database, in which traceability to the source of the requirement, and allocation to a given partition will be recorded.

Software Requirements Review

The objectives of the DCS software requirements review are to ensure that the requirements, as documented in the SRS, accurately reflect the needs of the process system, the Generic Requirements of the DCS hardware platform have been correctly interpreted, and any derived requirements resulting from the Software Requirements Definition activity are legitimate.

Software Detailed Design

After preparation of the SRS documents, the partition SRS requirements will be examined to determine the most appropriate allocation of each requirement to a specific control processor, either at the Group Control Level, or at the lower Device Control Level.

Additional 'derived' DCS requirements will be defined at this stage, which are DCS hardware or implementation specific rather than arising from plant system functional requirements. Examples of derived requirements would be self-checking and redundancy handling.

These derived requirements will be expressed in function block diagram format and entered in the Requirements Tracking and Verification database, where their traceability to the parent requirement will also be recorded. The result of the DCS detailed software design phase is the DCS software design description (SDD).

Software Design Review

The objectives of the DCS software design review are to ensure that the requirements contained in the SRS have been appropriately allocated to processors and software units, any requirements derived as a result of the detailed design process are legitimate, and the software architecture has been chosen to meet the performance requirements, and to facilitate subsequent test and maintenance activities.

Software Code Generation

Code Generation is largely an automated procedure, carried out using software development tools procured from the selected hardware vendor and, where necessary, developed by AECL.

The outputs of Code Generation are instruction list and configuration data files that are used to program the various modules in the DCS. These files determine the logic to be executed in each processor, the signals which will be transmitted on each communications link, and the frequency with which they will be transmitted.

Software Testing Activities

The objective of the software testing is to ensure the generated code correctly implements the software design and the specified software requirements. Software Testing will be addressed at three levels:

- Unit Testing, (the lowest level), in which the individual software units, and groups of units will be tested against the requirements allocated to them in the SDD;

- Subsystem Testing, in which each functional partition will be tested against the allocated SRS requirements; and
- System Integration Testing, (the highest level), in which any remaining SRS requirements not verified at the Subsystem Test level will be tested against the requirements in the SRS. This will, for example, verify requirements involving inter-partition communication.

DCS Application Software Validation

The PFS and DCR documents are used during the validation step (black box testing) of the DCS software as shown in Figure 2. Validation activities provide a check that the DID requirements were correctly interpreted by the software developers. Validation treats the DCS as a "black box". That is, the validator has no knowledge of the internal structure of the DCS (other than the partition divisions). In view of the complexity of the DCS, software validation will be conducted at two levels:

a) Partition Validation, which will be run in the software development facility, using the hardware and software for a single partition; and

b) System Validation, elements of which will be run at site on the *installed* system.

The CANDU 9 Plant Simulator, (described in Reference 3), will be used to perform the dynamic testing portion of the software validation activity. The simulator will be connected to the DCS via an interfacing subsystem so that the expected DCS parametric inputs (i.e. voltage signals) are provided.

Qualification of Pre-developed Software

The pre-developed software for the DCS controls the execution of function blocks, and performs various tasks associated with data communication and is the form of embedded (i.e. not user accessible) firmware in the various system components. For qualification purposes, this firmware is regarded as part of the DCS hardware.

Summary

The CANDU 9 plant design will use a distributed control system to perform plant monitoring and control functions previously implemented using digital control computers, analog control devices and relay logic. The design of the CANDU 9 DCS follows the proven methodology of previous computer control projects but supplements this process with new techniques and analyses. These new techniques will allow system designers to take advantage of new features possible in a DCS application, and ensure the DCS complies with all requirements of a modern nuclear power plant control system.

References

1) J.E. HARBER, M.K.KATTAN and M.J. MACBETH, "Distributed Control System for a CANDU 9 Nuclear Power Plant," 17th CNS Conference, June 19, 1996, Fredricton, New Brunswick.

2) COG-95-179, "Guideline for the Categorization of Software in Nuclear Power Plant Safety, Control, Monitoring and Testing Systems", September 1995.

3) M.K.KATTAN, M.J. MACBETH and K. LAM, "CANDU 9 Nuclear Power Plant Simulator," 19th CNS Simulation Symposium, October 1995, Hamilton, Ontario.

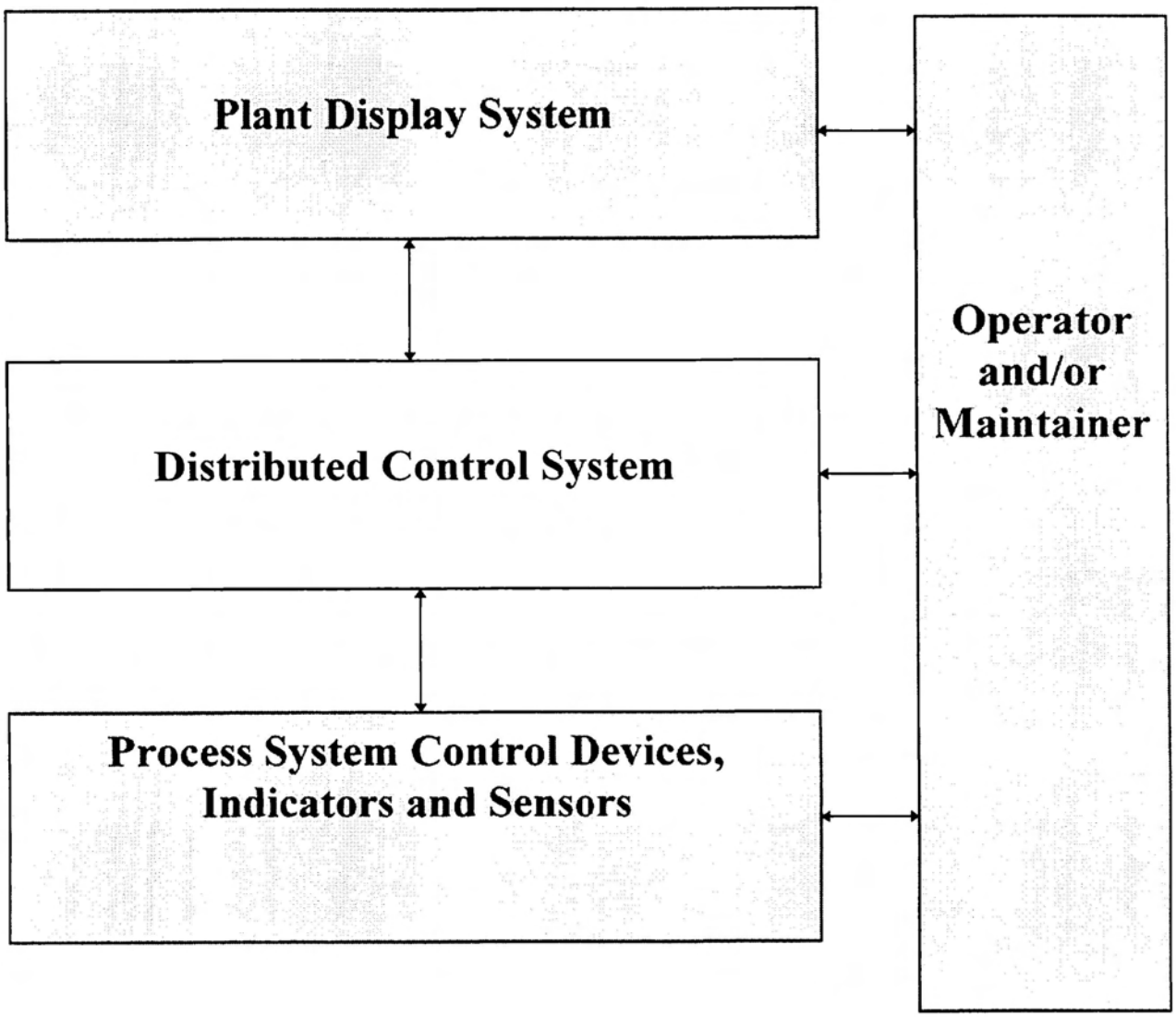

Figure 1 - Context Diagram of the CANDU 9 DCS

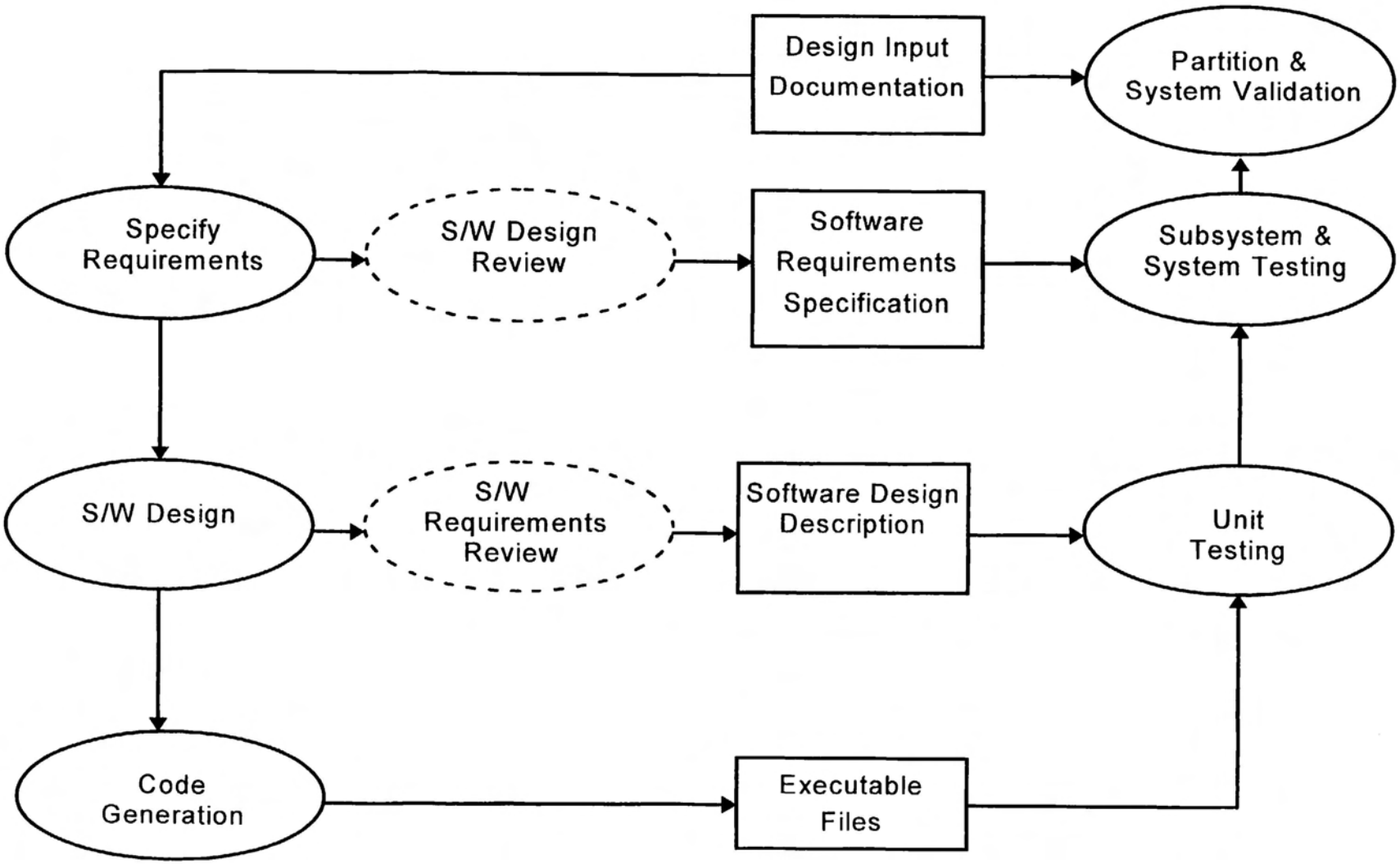

Figure 2 - Overview of the CANDU 9 Software Development Process for a Partition