원자력 발전용 플랜트 제어를 위한 로직 프로그램의 개발

김영춘*. 윤용한*. 김재철*. 황선주**. 이영길**. 박창두**
*: 숭실대학교 전기공학과 **: (주) 아시아계전 연구소

Development of Logic Program for Nuclear Power Plants Control

Young-Chun Kim*. Yong-Han Yoon*. Jae-Chul Kim*. Sun-Ju Hwang**. Young-Gil Lee**. Chang-Du Park**
*: Dept. of Elec.Eng Soong-Sil Univ. **: Asia Research & Development Co.

Abstract - This paper presents a basic interposing logic program to control nuclear power plant. In this paper we select a target control loop among the whole interposing modules, develop logic algorithm and functional software to compose target control loop. After that we carry out V&V(Verification and Validation) into the developed logic program to improve quality and reliability.

1. 서 론

원자력 발전소용 연동로직모듈(interposing logic modules)은 원자력 발전소에 소요되는 설비(펌프, 송풍기, 히터, 전동기구동 밸브, 솔레노이드 밸브, 전기 및 유체구동 밸브 등)를 자동으로 동작시키는 온-오프(on-off) 제어기로써 각종 입/출력 카드, 통신 카드, 제어카드 등의 주요 기기와 전원 공급 기기, 지시계, 기록계 등의 보조 기기로 구성된다[1]. 이러한 제어모듈은 루프의 종류가 매우 다양하므로 표준화된 기본모듈이 필요하다. 이러한 표준화된 기본모듈을 기능에 따라 적절히 조합하여 각각의 제어 기능을 수행하므로써, 연동모듈의 효율적인 제어 및 관리가 가능하다.

본 논문에서는 원자력 발전소의 플랜트 제어를 위한 로직 관련 기본 프로그램을 개발하였다. 이를 위해 연동모듈의 제어 루프중 일부를 선택하여 대상 제어루프로 선정하였고, 그에 따른 로직 알고리즘(logic algorithm)과 각 알고리즘의 구현을 위한 기능별 소프트웨어(functional software)를 구성하였다. 그리고 기존에 구성되어진 연동모듈의 FID(functional interconnection diagram)와 로직 알고리즘을 기반으로, 본 논문에서 개발한 로직 프로그램의 확인 및 검증(validation & verification)을 수행하므로써 연동 모듈 로직 제어기의 신뢰성을 향상시킬 수 있을 것이다.

2. 본 론

연동로직시스템은 공학적 안전설비계통의 많은 현장기기를 제어하고, 기기의 동작상태를 운전원, 발전소 컴퓨터 및 경보계통에 알려주는 시스템으로서 발전소의 안전성 측면에서 매우 중요하다[2]. 연동로직시스템에 사용되는 소프트웨어는 대부분 간단한 개별적인 특성을 갖는 모듈들로 구성되어 있으므로 신뢰성, 유지 보수성의 측면에서 큰 장점이 있다.

2.1 연동로직시스템 소프트웨어 구성

연동모듈의 각 루프를 제어하기 위해서는 기능별 소프트웨어가 필요하며, 이러한 기능들은 간단한 연산으로부터 복잡한 수치연산까지도 필요로 한다. 그러므로 실제 현장에 적용하기 전에 소프트웨어의 기능 검증이 필요하며, 최적인 제어 프로그램이 되도록 많은 경우의 모의시험을 통해 신뢰성을 향상시켜야 한다. 이를 위하여 연동로직시스템을 하향식 설계(Top-Down design) 방식으로 구성하였고, 연동모듈의 제어를 위한 소프트웨어는 크게 다섯 부분으로 나누어 구성하여 그림 1과 같은 내부 구성을 가지도록 하였다.

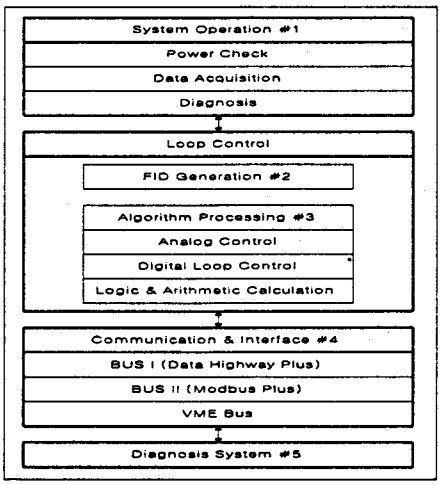

그림 1. 제어모듈의 구성도(loop/analog control)

로직 프로그램의 구현을 위해서는 대상 제어루프의 로직 알고리즘과 각 알고리즘의 구현을 위한 기능별 소프트웨어가 필요하므로 대상 루프제어를 위해 기본적으로 구성되어야 할 기능별 소프트웨어에 포함될 요소를 표 1과 같이 분류, 개발하였다.

표 1. 디지털 로직 및 아날로그 제어 알고리즘

Digital logic	Analog control algorithms
· Multiple Input AND & OR · Multiple Input Reflash OR · 2 Input EX-OR · Single Input Inverting Buffer · 2/3, 2/4, 3/4 Coincidence Matrix · Adjustable On/Off Delay · Flasher Logic Signal Generator · Memory Latch with Set/Reset Inputs · Toggle Memory Latch · Override Memory Latch · Adjustable Non-Retriggerable Pulse · Logic One/Zero · Up/Down Counter · Lamp Test Input	· Logic Control Algorithms · Arithmetic Algorithms · Comparison Algorithms · Linearization Conversion & Function Generation Algorithms · Value Selection and Limiter Algorithms · Dynamic Function Algorithms · Control Function Algorithms · Output Calculation Algorithms · Input / Output Processing Algorithms

2.1.1 제어모듈의 개발 과정

연동모듈을 구현하기 위한 소프트웨어는 각 기능별로 유용성의 검증과 불필요한 로직과 알고리즘의 존재여부에 대한 조사, 기존에 구성되어진 연동모듈의 FID와 로직 알고리즘을 기반으로 하여 개발된 소프트웨어의 확인 및 검증 작업이 필요하다. 그림 2는 이러한 일련의 과정을 나타내는 것으로 연동모듈 로직 제어기의 정상동작 유무 및 성능을 검증하여 시스템의 신뢰성을 향상시키기 위해 필요한 과정이다.

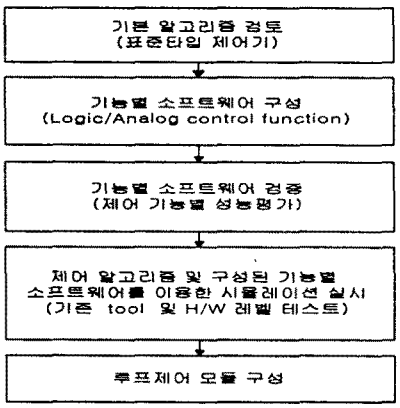

그림 2. 로직 제어모듈의 개발과정

2.1.2 루프 제어모듈 구성

실제 제어대상의 로직도면을 제어 알고리즘 모듈과 접속하기 위하여 상용화된 전자회로 설계프로그램인 OrCAD를 이용하여 제어모듈을 변환하였다. 기존의 P&ID도면을 OrCAD의 모듈형태 도면으로 작성하여, 각 모듈의 접속상태를 OrCAD의 netlist 파일 형태로 변환하였다. 변환된 netlist 파일은 C언어의 형태로 변환하기에 용이하게 이루어져 있으므로 이를 컴파일하여 메모리 영역에 복사하므로써 알고리즘 처리모듈과 접속을 이룰 수 있었다. 이러한 절차를 도표화하면 그림 3과 같다.

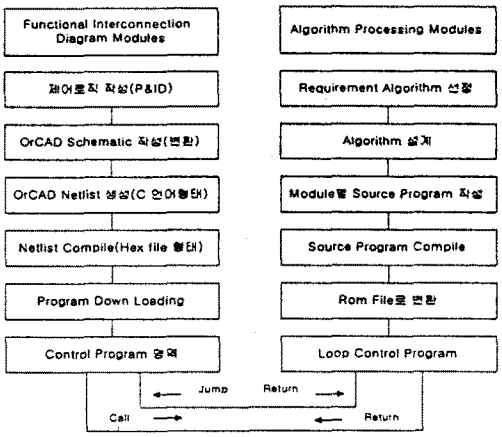

그림 3. 제어모듈의 처리절차

2.2 소프트웨어 설계 과정

2.2.1 소프트웨어 설계 요건

소프트웨어는 소프트웨어 품질보증 프로그램(Software Quality Assurance Program)을 작성하여 다루었으며 안전 관련 소프트웨어의 경우 SQAP는 10CFR50 부록B에 명시된 요건을 준수하였다. 소프트웨어 개발 단계에서는 요구 명세서, 기능 명세서, 상세한 소프트웨어 설계 명세서, 코딩 및 소프트웨어 생산, 시험, 설치 및 시운전(commissioning), 운영/보수 등을 포함하는 소프트웨어 수명주기를 수립하였다. 소프트웨어 확인 및 검증 계획(software verification and validation plan)을 작성하였으며 확인 및 검증 계획은 ANSI/IEEE Std 730, 829, 1012[4-6]등의 요건을 만족하도록 하였다.

2.2.2 소프트웨어의 설계

소프트웨어의 설계는 하향식 설계방식과 계층적 설계구조에 따라 일관된 모듈 설계원리를 이용하였고 복잡한 소프트웨어 구조를 피하였다. 소프트웨어의 설계에 있어서는 설계 기준 이상의 사건을 반영하였고 서로 다른 컴파일러, 운영체계, 프로그램언어 및 기타보조 소프트웨어 패키지 등의 사용은 최소화하였다. 소프트웨어의 설계는 알고리즘을 프로그램하는데 적합하고 널리 인정된 고급언어중 하나인 C 언어를 이용하여 작성하였으며 어셈블리 언어는 특별한 경우를 제외하고는 사용하지 않았다. 또한 계통의 기계적 요건에 무관하게 운용될 수 있도록 소프트웨어를 설계하였다.

2.3 소프트웨어 시험 검증

원자력 발전소 디지털 계측제어 소프트웨어 시스

템은 그 criticality의 정도에 따라 제어계통, 감시계통, 보호계통 소프트웨어와 그 외 여러가지 운전보조 시스템 소프트웨어 등으로 나누어지며 이들은 다시 안전관련 소프트웨어와 안전에 관련이 없는 소프트웨어로 나누어진다. 개발 단계에서 신중히 고려해야 할 부분은 일반적으로 안전과 직결되는 안전계통 소프트웨어의 품질과 신뢰성 보장 문제이다.

소프트웨어의 확인 및 검증이란 소프트웨어 시스템이 실제로 요구된 기능을 완벽하고 신뢰성 있게 수행함을 소프트웨어 개발공정 단계별로 확인하고 개발 절차에 따라 절차간에 정확한 정보교환이 되었는가를 검토하는 작업절차이다. 소프트웨어 검증은 옳바른 소프트웨어, 즉 목적하고 있는 소프트웨어를 구현하였는가에 대하여 분석하고 시험하는 작업으로 소프트웨어가 요구사항대로 설계되었음을 보장하는 절차이다[2].

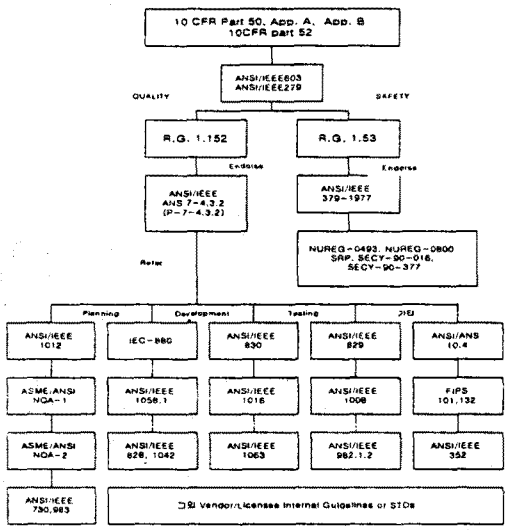

그림 4. 디지털 보호계통 소프트웨어 규제요건 체계

본 논문에서 확인 및 검증 작업은 프로그램의 각 모듈별로 문서 코드를 검토, 평가한 후 각각의 모듈이 소프트웨어 요건과 설계 및 코딩 기준에 일치하는지를 평가하는 작업으로서, 설계문서(design document)와 소스 코드에 대한 일차적인 검토가 끝난 후에 확인 및 검증 시험을 실시하였다. 확인 시험에 사용한 방법은 ANSI/IEEE Std.603-1980[3]에 의해 정의된 소프트웨어 모듈의 안전성 분류와 기능에 대하여 실시하였다.

확인 시험은 구조적 시험(structural testing)과 기능적 시험(functional testing)으로 나누어서 실시되었으며 이에 대한 기준안은 ANSI/IEEE 829-1983[5]에 나타나 있다. 구조적 시험은 모든 소스 프로그램이 디자인 명세에 맞게 작성되었음을 검토하는 것으로 본 논문에서는 이를 위해 소프트웨어 단일성과 복잡도(complexity)를 근거로 설정된 기준을 적용시켜 수행하였다.

확인 결과가 만족되면 목표 하드웨어에 소프트웨어가 설치되고 하드웨어와 소프트웨어가 검증 단계로 넘어간다. 검증 과정에서는 목표 하드웨어/소프트웨어에 대한 시스템 기능성(functionality)이 중점적으로 검토되는데, 검증 시험의 주요 단계는 다음과 같다.

1) "Top-Down" 기능적 요건 시험
2) 시스템 설계와 구현에 대한 비정상 조건 검토
3) MMI(Man-Machine Interface)시험

2.3.1 확인 및 검증 프로그램

확인 및 검증 프로그램에는 개발 단계에서부터 확인 및 검증 단계의 각 단계에서 필요한 검증 활동을 문서화하였다. 개발용, 검증용 및 확인용 도구는 상호 독립적인 것으로 하였고 각 도구의 사양은 확인 및 검증 프로그램에서 결정되도록 하였다. 검증 및 확인 작업은 설계와는 독립된 팀에 의해 수행되었다.

3. 결 론

원자력 사업에 있어서 아날로그 기기의 노후화 문제를 해결하고, 원전의 신뢰성과 가용성을 높일 수 있는 방법은 디지털 기술의 적용이다.

본 논문에서는 원자력 발전소의 플랜트 제어를 위한 로직 관련 기본 프로그램을 개발하였다. 이를 위해 대상 제어루프를 선정하여 그에 따른 로직 알고리즘과 각 알고리즘의 구현을 위한 기능별 소프트웨어를 구성하였다. 그리고 기존에 구성되어진 연동모듈의 FID와 로직 알고리즘을 기반으로 본 논문에서 개발한 로직 프로그램의 확인 및 검증 작업을 수행하여 원자력 발전용 플랜트 시스템의 신뢰성을 향상시킬 수 있을 것으로 생각된다.

원자력 사업에서의 계측제어 시스템 디지털화와 이를 위한 신뢰도 높은 소프트웨어의 개발을 위해서는 소프트웨어 확인 및 검증 기술의 적용이 필요한 사항이며 우리 실정에 맞는 개발, 규제요건, 규제방법 등에 대한 구체적 규정이 필요하다는 결론을 얻었다.

[참 고 문 헌]

[1] 박창두 외, "원자력 발전소용 연동모듈 개발", 아시아 계전(주), p.p. 130-136, 1997.

[2] 권기춘 외, KAERI/AR-411/94 "원전 계측제어 고신뢰도 소프트웨어 확인/검증기술현황", 한국원자력연구소, Chap 5, 1994.

[3] IEEE/ANSI 603-1980 "Standard Criteria for Safety System for Nuclear Power Generating Stations", IEEE, 1980.

[4] IEEE/ANSI 730.1-1989 "Standard for Quality Assurance Plans", IEEE, 1989.

[5] IEEE/ANSI 829-1983 "Standard for Software Test Documentation", IEEE, 1983.

[6] IEEE/ANSI 1012-1986 "Standard Criteria for Software Verification and Validation Plans", IEEE, 1986.