

운전원훈련 시뮬레이터용 디지털 제어시스템 번역기 개발

마삼선, 김종현

한국전력공사 전력연구원, 한국중공업(주) 기술연구원

Development of Digital Control System Translator for Operator Training Simulator

Sam-Sun Ma, Jong-Hyun Kim

Korea Electric Power Corporation, Korea Heavy Industries & Construction Co.

Abstract

발전소 운전원 훈련용 시뮬레이터 개발시 제어로직과 MMI(Man Machine Interface)의 화면개발은 많은 인력과 시간이 수반되는 일이지만 일일이 코딩을 하여만 실제 제어기능을 구현할 수 있다. 하동화력발전소 운전원 훈련용 시뮬레이터에서는 제어시스템 개발의 생산성과 신뢰도를 향상시키기 위해 번역기를 개발하여 현장에서 사용중인 제어파일을 일시에 번역하여 시뮬레이터에 적용하는 방법을 적용하고 있다. 이 논문에서는 하동화력 DCS인 Siemens Control에 적합한 제어 번역기의 개발과정을 서술한다. 번역기를 통한 제어시스템의 모델링 방법은 첨단제어 시스템의 기존 기능을 그대로 유지할 수가 있게 되어 제어기술자의 훈련 및 제어로직 개발 그리고 분석이 가능하며, 이를 통하여 운전원 훈련효과를 제고할 수 있고 실제 제어시스템과 완벽한 호환성을 유지함으로써 시뮬레이터의 정확성을 보증할 수 있게 된다.

1. 개 요

발전소 운전원 훈련용 시뮬레이터의 개발은 크게 세 부분으로 나눌 수 있다. 실 플랜트를 모델화한 Process model, 제어시스템을 모델화한 Control model 그리고 발전소 주 제어실에 있는 각종 Panel들의 모델로 나뉘어 진다. 일반적으로 Panel들은 Monitoring 및 조작 스위치들로 구성되어 있어 실 플랜트와 동일하게 제작되어 단지 I/O interface를 이용하여 Process 및 Control model과 연계하여 동작하게 된다.

일반적으로 Process 및 Control model은 시뮬레이션을 위하여 개발된 특정 S/W를 사용하여 개발하게 된다. 그러나 Control model의 경우 컴

퓨터 기술의 발달로 인하여 실 플랜트의 제어시스템들이 고도화 되어 각 제어 알고리즘들이 S/W로 저장되어 있어, Model 개발시 이들 알고리즘을 포함한 S/W를 시뮬레이션용 S/W에서 사용가능토록 하는 변환기술을 이용할 수 있다. 본 논문에서는 하동 화력발전소 DCS인 Siemens Control System에 적합한 DCS 변환기의 개발에 관하여 설명한다.

시뮬레이터에서 제어시스템 변환기를 사용하여 control model을 개발하였을 경우의 장점으로는 실제 제어 시스템 동일한 기능을 발휘하며, 실제 제어 시스템이 변경에 빠르게 대처할 수 있다. 또한 제어 기술자들이 시뮬레이션용 S/W가 아닌 기존의 제어 로직 개발에 사용되고 있는 시스

템을 이용하여 제어 로직을 개발할 수 있어 제어 기술자의 훈련 및 제어 로직 검증등에 적극 활용할 수 있다.

2. 본 론

2.1 Simulator 구성

하동 화력발전소는 한국 표준형 500MW 석탄 화력발전소로서 관류형 초임계압 보일러로 설계되어 있다. 먼저 시뮬레이터의 하드웨어 구성을 그림 1.에 나타내었으며, 주요 구성요소들은 다음과 같이 몇 개의 그룹으로 나뉘어 진다.

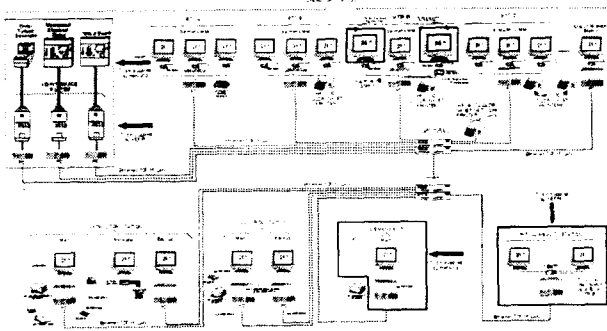


그림 1. Simulator Hardware Schematic

- Operator Station
 - Hard Panel
 - Siemens MMI
 - Mark V MMI
 - Vibration Monitoring System
- Instructor Station
 - Process Model
 - Instuctor Program
- Control Station
 - Siemens Control Model
 - Mark V Control Model
- Siemens EWS
 - Original Siemens Control Logic

- Virtual Reality Station

하동 시뮬레이터는 Dynamic modeling simulation tool인 ProTRAX로 이용하여 개발 되었으며, 프로세서 모델은 ProTRAX system중 GUI를 지원하는 GCPARM Program으로 개발하였다. 그리고 발전소 전체의 주 제어 시스템인 Siemens DCS와 터빈제어기인 Mark V의 경우는 제어 로직 변환기를 사용하였다. 그림 2.에 시뮬레이터 소프트웨어 구성을 나타내었다.

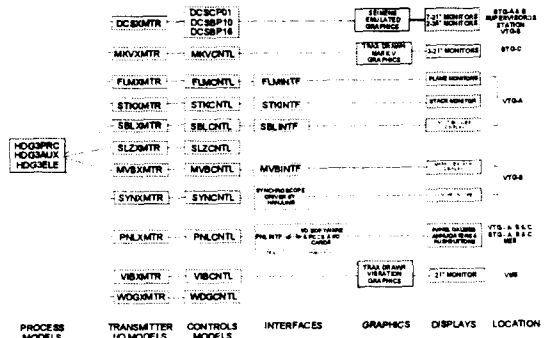


그림 2. Simulator Software Schematic

- Plant Process Model
 - HDG3PRC : BLR & BOP System
 - HDG3AUX : AUX System
 - HDG3ELE : Electrical System
- Transmitter I/O Model
 - Software I/O Interface
 - Transmitter
- Control Model
 - Siemens DCS System
 - GE Mark V System
 - Sulzer & Woodward Control System
 - Sootblower & Vibration System
- Interface
 - Hardware I/O Interface System
- Graphics
 - Siemens DCS MMI

- GE Mark V MMI

2.2 Siemens DCS Translator 개발

가. Siemens DCS Teleperm ME

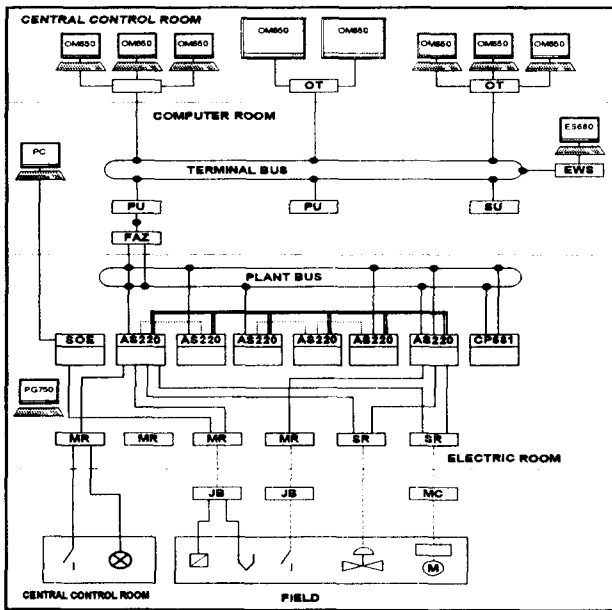


그림 3. Communication relationship of Teleperm ME

그림 3은 하동 화력 발전소의 주 제어 시스템으로 보일러 및 BOP 계통을 제어하는 Siemens Teleperm ME DCS의 구성도를 나타내었다. 주요 구성 시스템과 그 기능은 다음과 같다.

- OM650 (공정제어 및 정보처리 시스템)
 운용자와 프로세스를 연결하는 시스템으로서 운용자에게 프로세스 제어, 감시 및 정보처리 기능 등의 종합관리를 할 수 있는 역할을 한다.
- ES680 (엔지니어링 시스템)
 Teleperm DCS의 엔지니어링, 상세 설계 및 프로그램 구축등에 사용되며, 기능으로는 기기 기본 및 장치 설계, 그래픽 및 제어 로직 설계 그리고

데이터베이스 구축등이 있다.

- AS220 (자동제어 시스템)
 그룹 및 개별 계층의 자동제어를 수행하며 프로세스로부터 측정된 측정치와 상태 변수의 획득, 프로그램된 Open & Closed Loop의 기능에 따른 연산 제어 그리고 운용자의 조작내용을 프로세서에 전달하는 역할을 한다.

나. Translation Process

Siemens Teleperm ME Translator (TME_Tran)는 ES680에 저장된 제어 로직(Functional logic Diagram)을 시뮬레이터 개발용 Tool인 ProTRAX에서 사용 가능한 제어모델로 변환하는 작업을 수행한다. Translation을 위해서는 먼저 ES680의 HP-UX 시스템에 저장된 Ingres database와 control definition files에 접근하여 제어 로직에 관한 정보를 획득해야 한다. 그림 4는 Translator를 위한 정보 흐름을 나타내었다.

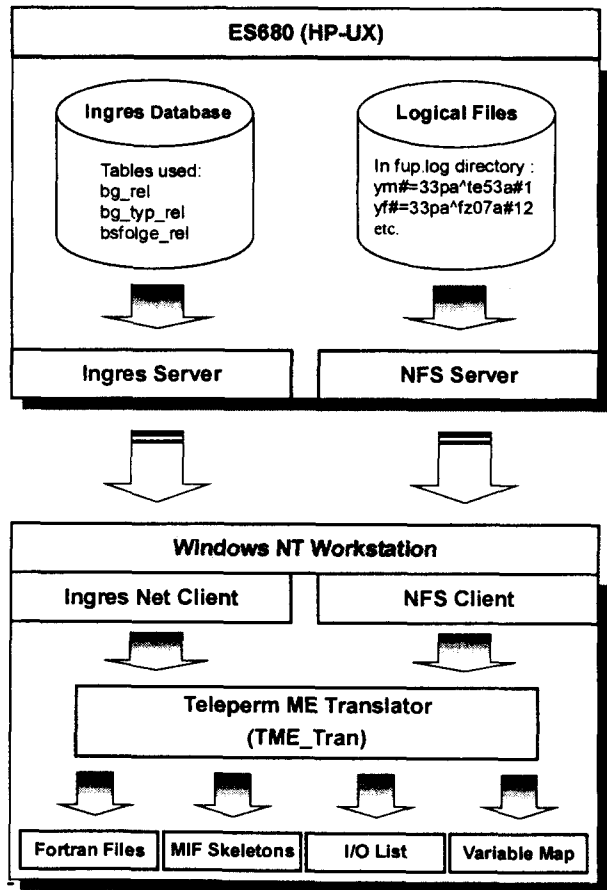


그림 4. Overview of the Data flow Translator

- Access to the Ingres database
ES680내의 Ingres database로의 연결은 embedded SQL programming tool과 Ingres/Net client for Windows NT를 이용한다. Database로 부터 다음의 정보들을 획득한다.

- 제어 로직 설계에 이용되는 모듈들의 List
- 모듈 Type Information
- 모듈 실행 순서

- Access to the Controls Definition Files
ES680내의 제어 관련 파일을 읽기 위해서는 NFS Maestro Solo를 이용하여 network driver로 연결한다. Control definition file로부터 functional diagram graphic 정보와 logical 정보를 획득한다.

다. Translator Configuration File

Translation 수행을 위해서는 configuration file을 필요로 한다. 이 file에는 translation될 source file의 위치와 생성될 file의 이름등을 정의하게 된다. 다음에 전형적인 configuration file을 나타내었다.

```

[PROJECT DIRECTORY]
k:\pro\h34get

[DBMS SERVER]
h3hp

[OUTPUT FILE NAME]
0cp01

+ [FORTRAN SUBROUTINES]
2
    
```

이 file은 운용자에 의해서 작성되며, 각 구문의 정의는 다음과 같다.

- [PROJECT DIRECTORY] entry
이 구문은 network driver로 연결된 ES680 시스템의 project name directory를 정의한다. 이 directory가 translation될 file들을 포함한다.
- [DBMS SERVER] entry
이 구문은 Ingres database file이 저장되어 있는 ES680내의 directory를 정의한다.
- [OUTPUT FILE NAME] entry
이 구문은 translation된 후 생성되는 file name의 기본이 된다.
- [FORTRAN SUBROUTINES] entry
이 구문은 translation된 후 생성되는 fortran file들의 수를 정의한다.

라. Translation Operation

Translation 실행을 위해서는 다음과 같은 명령어를 사용한다.

```
TME-Tran<config_file><AWE_name>[-m][-v]
```

여기서,

- <config_file>은 translator 실행시 사용되는 configuration file의 이름이다.
- <AWE_name>은 translate될 AWE의 이름이다. 이 file name은 연결된 ES680에서 필요한 파일에 대한 경로를 나타낸다.
- -m 이 option은 AWE-to-AWE MIF file을 생성하게 한다.
- -v 이 option은 ProTRAX variable map과 cross-reference file을 생성하게 한다.

그림 5.는 translation이 실행되면 화면을 나타내었다. 그리고 Translator는 다음과 같은 file들을 생성한다.

- COMPILE.FOR
Fortran source file로 MAIN program unit, model

의 정의 그리고 constant operand의 초기화등을 포함한다.

- COMPILE.INC

Fortran include file로 model내의 모든 변수 선언과 model COMMON blocks 정의를 포함한다.

- MODEL1.FOR

표준 ProTRAX model내의 flow model에 연결되는 Fortran routine

- MODEL2.FOR

표준 ProTRAX model내의 energy model에 연결되는 Fortran routine

- CNTRLn.FOR (n == 1...9)

Translate된 control coding을 포함하는 subroutine

- <base>.CMI

AWE와 AWE사이의 변수 interface를 정의한다.

- <base>.MIN

Control model이 필요로 하는 process로 부터의 입력변수를 정의한다.

- <base>.MOT

Control model로 부터 출력되는 process 변수를 정의한다.

- <base>.VMP

변수 map과 변수 cross-reference를 제공한다.

용하므로 정확성과 호환성이 보장된다. 최근의 발전소 시뮬레이터는 운전원 훈련뿐만 아니라 발전소 제어로직개발등에도 이용될 수 있는 다목적 시뮬레이터가 요구되고 있는 실정이다. 이러한 요구를 만족하기 위해서는 제어로직 변환기술이 필수적이라 할 수 있다. 또한 이러한 제어로직 변환기술은 제어 로직 개발 및 검증등 다양한 활용이 가능하다.

3. 결론

제어시스템의 emulation 방법으로는 실제 제어 로직의 code나 도면을 참고로 하여 modeling 하는 방법과 실 제어로직 source code를 제어 시스템 변환기를 통하여 변환하는 두가지 방법이 있다. 전자의 경우 범용제어이론을 이용하여 구현되기 때문에 정확성 및 실제 제어 시스템과의 호환성을 보증하지 못한다. 그러나, 후자의 경우 실제 제어 시스템에 사용되는 제어로직을 변화하여 사