

시뮬레이터 디스플레이 및 제어 시스템 개발

서재승* · 김성배

고등기술연구원

요약

고등기술연구원에서 개발한 중수로형 발전소 시뮬레이션 프로그램을 적용하여 시뮬레이션된 결과를 그래프 및 그래픽 객체를 이용하여 모니터에 디스플레이하며, 그래픽 사용자 인터페이스(GUI) 개념을 이용한 사용자와 시뮬레이터 프로그램 사이의 실시간 상호 정보 교환을 위한 프로그램을 X-윈도우 환경에 기초를 두고 있는 SCADA 개념의 그래픽 데이터 프리젠테이션 틀을 이용하여 개발하였다. 시뮬레이터 디스플레이 시스템은 X-윈도우의 그래픽 라이브러리에서 제공하는 기능을 이용하여 사용자에게 시뮬레이션 결과의 정보를 전달하며, 단지 결과를 보는 것에 비하여 사용자의 제어 값을 입력하여 모사를 할 수 있도록 기계와 사용자 사이의 인터페이스를 제공하는 제어 입력 시스템을 개발하고 이를 이용하여 발전소 운전원의 교육 및 발전소 주요 시스템의 운전 상태와 운전 변수들의 감시 및 제어 기능을 제공하도록 설계하였다.

1. 서론

원자력 발전소의 디스플레이 시스템은 발전소에서 생성되는 모든 값을 측정하고 디스플레이를 이용한 모니터링 및 컴퓨터를 이용한 사용자의 제어 기능을 모두 갖추고 있다. 시뮬레이터 디스플레이 시스템은 중수로형 원자력 발전소의 디스플레이 시스템에 준하여 발전소의 측정값을 중수로형 원자력 발전소 시뮬레이션 프로그램의 계산 값으로 대치한 모니터링 및 사용자의 제어 기능을 갖춘 디스플레이 시스템이다.

1.1 네트워크 중심의 컴퓨터 환경

지금까지는 대부분의 컴퓨터들이 독립된 환경 속에서 자라 왔지만, NOS¹⁾는 컴퓨터들을 하나의 끈으로 묶어 줌으로써 하나의 컴퓨터 사회를 이룰 수 있게 되었다. 물론 이미 널리 퍼져 있는 네트워크 환경이 컴퓨터 사회를 만들고 있지만, 여기에는 몇 가지 문제점을 내포하고 있다. 지금까지 네트워크란 컴퓨터를 잘 아는 일부 컴퓨터 매니아들의 소유물이었다. 개인용 PC에서 네트워크를 사용하려면 도스²⁾ 명령 말고도 네트워크 서버가 제공하

는 명령들을 익혀야만 했다. 또한 각 네트워크 드라이버들은 도스의 기본 메모리 공간에 위치함으로써 도스의 메모리 영역이 줄어들어 여러 개의 네트워크 클라이언트를 동시에 띄우는 것을 힘들게 해 왔던 것이 사실이다.

네트워크란 동일한 작업을 둘 이상의 컴퓨터에서 처리하거나 프린터 등의 시스템 자원을 공유할 경우 사용된다. 네트워크에서는 크게 두 가지의 연결 방법이 있다. 클라이언트-서버 방식과 피어 투 피어(Peer to Peer)방식이 바로 그것이다. 클라이언트-서버 방식은 한대의 서버 컴퓨터에 여러 대의 클라이언트 컴퓨터가 연결된 방식으로 각각의 컴퓨터에 네트워크 카드 구동 프로그램을 띄워 주면 상대방 컴퓨터와 컴퓨터에 연결된 주변기기가 자신의 컴퓨터에 연결되어 모든 컴퓨터 사용자들이 서버의 시스템 자원에 접속할 수 있다.

디스플레이 시스템은 LAN을 기반으로 하는 서버-클라이언트 방식의 네트워크를 구성하였다. 서버인 워크스테이션에서는 시뮬레이션 프로그램 및 그래픽 사용자 인터페이스용 프로그램을 공유 메모리 기법을 이용하여 시뮬레이션의 결과를 그래픽 사용자 인터페이스

1. Network Operation System

2. Microsoft Disk Operating System

용 프로그램에 전달하는 방법으로 실행하며, 시뮬레이션 프로그램의 결과를 클라이언트 PC에서 디스플레이하고 모니터링 및 제어할 수 있도록 X-터미널 환경을 클라이언트에서 사용하였다. 공유 메모리 기법을 사용하므로서 시뮬레이터 디스플레이 시스템은 실시간의 시뮬레이션을 어느 프로세서에서든지 같은 시간대의 결과를 디스플레이할 수 있으며, 특히 네트워크를 통하여 디스플레이 되는 각 시스템의 도면에서 변수 값의 일률적인 Update가 가능하다.

1.2 하드웨어와 소프트웨어의 구성

시뮬레이터 디스플레이 시스템과 제어 입력 시스템은 LAN을 통한 하나의 동일한 네트워크 환경에 설치하였으며, 각 시스템은 크게 하드웨어와 소프트웨어로 구성된다. 그림 1은 하드웨어의 평면적인 구성도를 나타내고 있으며, 서버와 X-터미널로 사용될 PC가 클라이언트로 연결되었으며, NOS를 NFS PC에 설치하여 전체 네트워크를 제어한다. 표 1은 사용된 소프트웨어의 목록으로 그래픽 개발 도구로 DataViews를 사용한다.

표 1. 소프트웨어 목록

구분	목록
서버	HP UNIX 9.03A
	X11R5 X-Window
	HP C Compiler
	HP FORTRAN77 Compiler
	HP Softbench
	DataViews
클라이언트	MS-DOS 6.22
	MS-Windows 3.1
	MS-Windows NT 3.5
	Future TCP/IP
	MS LAN MAN System
	Exceed for Windows

2. 그래픽 사용자 인터페이스 개발

그래픽 개발 도구의 하나인 DataViews는 이차원 그래픽 사용자 인터페이스를 개발하는 환경을 제공하고, 실시간 감시 및 제어 화면을 구성하며, 그래픽 개발 도구는 개방형 구조를 사용하여 그래픽 사용자 인터페이스를 개발하므로, 높은 이식성을 가지고 있다. 이

러한 그래픽 개발 도구는 Man-Machine 인터페이스의 구성에 있어 그래픽 도면 편집기를 통한 프로그램 작성의 용이성 및 시뮬레이션을 통한 사용자 인터페이스 환경 개발을 제공하는 범용 도구이다.

2.1 그래픽 개발 도구의 구성

그래픽 개발 도구는 기본적인 그래픽 도면 편집기와 X-윈도우용 그래픽 라이브러리로 구성되어 있다. 그래픽 도면 편집기를 이용하여 사용자 인터페이스 화면인 View 도면을 구성할 수 있으며, 이는 X-윈도우 환경에서 사용자가 윈도우 및 그래픽 객체 설정 프로그램을 작성하지 않아도 되며, 구성된 View 도면을 취합, 제어하는 독자적인 프로그램을 구성할 수 있다.

2.2 그래픽 개발 도구의 기능

1) 그래픽 도면 편집기

그래픽 도면 편집기는 응용 프로그램의 사용자 인터페이스를 구성 가능케 하여 개발자에게는 "Point and Click"을 통한 "Menu Driven"을 이용, 화면 구성에 있어 "Low Level"의 프로그램 작성 작업을 최소화 시켜 주며 사용자 인터페이스의 기능을 최대화 할 수 있는 기능을 제공한다. 또한 사용자 인터페이스를 상호 연동 하여 프로토타입을 구성하도록 지원을 하며 다음 기능을 갖는다.

가) 그래픽적인 객체

직선, 타원, 호, 다각형, 아이콘, 그래픽 도면 같은 기본적인 객체와 막대 차트, 선 그래프, 계측기 같은 그래프 및 다양한 입력 객체를 그래픽적인 객체로 지원한다.

나) 다이내믹 선택사항

파일, 사용자 입력, 사용자 함수 등의 다양한 종류의 데이터를 그래픽적인 객체들과 연계하도록 지원하며, 기본 제공되는 그래프에 실시간 데이터를 연계하여 다이내믹한 표현이 가능하도록 지원한다. 또한 그래픽적인 객체에 연계된 데이터 근원의 값에 따라 View 화면의 속성을 변화시켜 주는 기능도 제공한다.

2) X-윈도우용 그래픽 라이브러리

그래픽 도면 편집기를 통해 구성된 화면을 통합하여 독립된 응용 프로그램으로 구성토록 지원하는 C 언어로 구성된 부프로그램적인 라이브러리이며, 이를 통해 다른 응용 프로그램들이나 다이내믹 데이터 근원들, 특정 시스템, 데이터베이스 등과 연결하여 사용자 인터페이스를 구현할 수 있도록 지원한다.

3) 그래픽 개발 도구의 Block 구성도

DataViews의 구조는 View 도면을 중심으로 View 도면을 디스플레이 할 수 있는 디스플레이 장치 및 X-윈도우 환경의 윈도우와 View 도면에 정의된 그래픽적인 객체들과 객체에 데이터를 전달하는 데이터 근원 등으로 구성되어 있다. 특히 그래픽 객체와 데이터들 사이에는 다이내믹한 연결을 구성할 수 있는데 이는 View 도면에서 데이터의 Update시 각 그래픽적인 객체도 다이내믹하게 Update될 수 있는 기능을 제공한다. 그림 2는 Data Views의 Block 구성도이다.

2.3 View 도면 구성

그래픽 도면 편집기를 통해 작성된 View도면의 각 그래픽 객체의 배치와 그래픽 데이터의 디스플레이와 그래픽 객체의 구성 등은 기본적으로 원형 원자력 발전소 1호기의 주 제어실의 디스플레이에 기초를 두고 구성하였다.

1) View 도면의 변수

View 도면의 입력용 변수는 표 2에 나타나 있으며 각 변수의 기능은 변수내용에 설명되어 있다.

표 2. View도면의 입력용 변수

View	변수	변수 내용
Control Panel	Rx Power	원자로 출력
	Trip	원자로 Trip Flag
	Step Back	Power Step Back
	Set Back	Power Set Back
	Shutdown	원자로 Shutdown Flag
	Restart	프로그램 재실행
	Quit	프로그램 종료

View 도면의 출력용 변수는 표 3에 나타나 있다. 시뮬레이션 프로그램에서 생성된 데이터와 View 도면의 변수와의 다이내믹 연결을

위하여 데이터 테이블을 각 View마다 만들어 주어야 한다.

표 3. View도면의 출력용 변수

변수	View 도면	변수 내용
압력	원자로	원자로 입/출력 압력
	증기발생기	증기발생기 입/출력 압력
	가압기	가압기 압력
	PHTS	각 노드별 압력
온도	원자로	원자로 입/출력 온도
	증기발생기	증기발생기 입/출력 온도
	가압기	가압기 유체 온도
	PHTS	각 노드별 온도
기타	원자로	원자로 출력, Zone level, Controller Status
	증기발생기	Water level, Feed/Steam Flow Rate
	가압기	Water level, Feed/Bleed Flow Rate
	PHTS	각 노드별 Flow Rate

2) 시스템의 View 도면

각 시스템의 View도면은 View도면에서 입/출력되는 변수와 기능에 의해서 발전소 전체 View 도면, 원자로 View 도면, 증기발생기 View 도면, 가압기 View 도면, 일차 열전달 계통 View 도면, 감속재 온도 제어 View 도면, 발전기 View 도면과 주 제어 패널 View 도면으로 나눌수 있다. 그림 3은 View 도면의 예를 나타내고 있으며, 표 4는 View의 종류별 분류 표이다.

표 4. View도면의 종류별 분류

시스템	View 도면
전체 View	Introduction, View 도면 Schematic, CANDU NPP Schematic
원자로 View	Reactor Core Status, Adjuster Rod Status, RRS, Zone Deviation, Reactivity Control Diagram, Stepback Status, Setback Status, UPR Status
S/G View	S/G Status, BPC, BLC Status, Boiler System Schematic
가압기 View	Pressurizer Status
PHTS View	PHT Schematic, PHT Control, Plant Schematic
Control Panel	Simulation Control Panel

3. 그래픽 사용자 인터페이스 프로그램

디스플레이 시스템의 구성은 사용자가 그래픽 도면 편집기를 이용하여 그래프 및 그래픽 객체를 포함한 View도면, View 도면에서 디스플레이할 데이터를 생성하는 응용 프로그램과 응용 프로그램에서 데이터를 받아서 View 도면에 디스플레이 하는 인터페이스 프로그램으로 나눌 수 있다. 특히 FORTRAN으로 작성된 응용프로그램과 C로 작성된 그래픽 사용자 인터페이스 프로그램 사이에는 데이터 공유를 위한 프로그램이 필요하며 이는 공유 메모리 기법을 이용하여 작성했다.

3.1 사용자 인터페이스의 규칙

사용자 인터페이스란 응용 프로그램의 사용자에 대한 그래픽적인 표현으로, 응용 프로그램에서 계산되어 전달된 정보와 응용 프로그램의 상태(status)를 디스플레이하며, 또한 사용자가 응용 프로그램을 제어할 수 있는 기능을 제공한다. 인터페이스의 동작은 그림 4에 나타낼 수 있으며, 각 기능은 다음과 같다.

1) Application => Interface

인터페이스는 응용 프로그램의 데이터 형태로 응용 프로그램에서 입력을 받는다.

2) Interface => User

인터페이스는 그래픽적인 데이터 디스플레이 형태로 사용자에게 출력을 전달한다.

3) User => Interface

인터페이스는 키보드에서의 입력과 마우스에서의 입력과 다른 입력 주변기기에서의 입력 형태를 사용자로부터 받는다.

4) Interface => Application

인터페이스는 응용 프로그램의 제어 형태로 응용 프로그램에 출력을 전달한다.

3.2 인터페이스 프로그램의 작성

그림 5는 인터페이스 프로그램의 제어 흐름도를 나타내며, 기본적인 인터페이스 프로그램의 체계적인 구조를 나타낸다. 그림 5에서의 흐름도에 따라서 인터페이스 프로그램의 주 프로그램을 작성하며 주 프로그램의 구성별 내용은 다음과 같다.

1) 변수 및 사용자 함수 선언

인터페이스 프로그램에서 사용될 모든 변수 및 사용자 함수를 선언해야 한다. 응용 프로그램과 인터페이스 프로그램 사이의 상호 데이터 공유를 위한 변수는 구조체 형태로 선언을 한다. X-윈도우용 라이브러리 환경을 사용할 수 있도록 해더 파일을 주 프로그램에 포함시킨다.

2) 인터페이스 프로그램의 초기화

인터페이스 프로그램에서 사용될 모든 변수 및 사용자 함수가 선언되면, X-윈도우의 그래픽 라이브러리를 사용하는 인터페이스 프로그램의 각 변수 및 함수를 초기화를 해야 한다. 기존에 동작중인 윈도우를 닫고, 새로운 윈도우를 열 수 있도록 각 윈도우의 변수 값을 초기화를 시킨다.

3) 모든 View 도면의 탑재

그래픽 도면 편집기에서 작성한 각 시스템의 View 도면을 불러들인다. 응용 프로그램에서 생성된 데이터와 화면에 보여줄 각각의 View 도면의 변수를 다이내믹하게 연결하기 위하여 각 View 도면의 심벌 테이블을 작성한다.

4) 윈도우 생성

View 도면을 디스플레이할 윈도우를 지정하고 생성시킨다. 인터페이스 프로그램에서는 모든 정보를 윈도우의 View 도면을 통해서 사용자에게 전달하고, 윈도우의 입력용 그래픽 객체를 통해서 제어 값을 입력받는다.

5) 시뮬레이션의 시작

인터페이스 프로그램에서 응용프로그램을 실행시켜 View 도면에서 디스플레이할 데이터를 계산할 수 있도록 한다.

6) View 도면에서 커서의 입력값을 초기화

모든 윈도우의 View 도면에 데이터를 디스플레이 하면서 사용자의 제어 값을 받아들이기 위한 커서의 입력값을 초기화한다.

7) 종료하기 전까지 무한 루프

이벤트 처리 루틴에서 종료 플래그 값을 입력받을 때까지 전체 인터페이스 프로그램을 실행하면서 각 View 도면의 변수 값을 응용

프로그램의 데이터로 Update를 한다.

8) 각 윈도우에서의 사용자 입력 처리
각 윈도우에서의 사용자가 입력한 플래그 값을 인식하여 각 입력된 플래그 값에 대한 이벤트를 처리한다.

9) 인터페이스 프로그램의 종료
응용 프로그램의 실행이 끝나면, 모든 View 도면을 위한 윈도우를 닫고 인터페이스 프로그램을 종료시킨다.

4. 결론 및 논의

시뮬레이션 디스플레이 시스템의 각 디스플레이 도면은 월성 원자력 발전소 1호기의 주 제어실의 디스플레이 도면을 참조하여 작성되었으며, 자체 설계한 디스플레이 도면을 탑재하여 모니터링 및 제어기능을 시험하였다. 기존 도면에 비해 그래픽적인 재계의 첨가와 탁월한 화면의 처리로 모니터링 기능이 향상되었지만, 응용 프로그램과의 제한된 연계로 부분적인 사용자의 제어 기능만을 수행할 수 있다. 사용자의 제어 값을 입력하는 주변기기로 터치 스크린을 사용하여 아날로그 센서에 의한 제어값의 입력을 시도하여 사용자와 그래픽 사용자 인터페이스 프로그램 사이의 상호작용에 편의성을 제공했다.

시뮬레이션 디스플레이 시스템의 전체적인 구성은 서버와 클라이언트가 서로 다른 독립된 운영체제를 사용하며, 단지 LAN을 이용하여 상호 데이터를 교환하고 있다. 즉 클라이언트는 서버의 터미널 형태로 존재하므로서 서버의 모든 기능을 동시에 클라이언트에서 실행할 수 있는 장점이 있지만 서버의 연산처리에 과중한 부하를 줄 수가 있으며, 이는 클라이언트가 서버의 터미널 개념이 아닌 독립된 하나의 서버의 기능을 수행할 수 있는 개방형 시스템을 구성하여 각 프로세서를 분산시켜서 서버의 과부하를 해결 해야한다.

5. 감사

본 연구과제는 과기처 시행 '94 원자력 연구개발 사업의 일환으로 채택되어 원자력연구소의 주관(CO94-01)으로 "CANDU형 발전

시스템 설계·운전 지원용 플랜트-제어 모사 해석기 구현"의 1차년도 사업으로 수행되었기에 이에 감사드리며, 한국원자력연구소 장종화 박사님의 공유메모리에 관한 조언에 감사드립니다.

참고문헌

1. "DV-Draw User's Guide", V.I Co.
2. "DV-Tools User's Guide", V.I Co.
3. "DV-Tools Reference", V.I Co.
4. "HP FORTRAN/9000 Programmer's Reference", Hewlett Packard Co. (1992)
5. "HP-UX Reference Vol. 2", Hewlett Packard Co. (1992)
6. The DSNP Group-UNB, 1992, "DSNP Users Guide-How to Write a DSNP Program from Start to Finish-Revision 1.02", Edited by J. Pitre, University of New Brunswick.
7. 김동수 외, "Nuclear Plant Analyzer 개발", KAERI/RR-902/89
8. "퍼스널 컴퓨터를 이용한 훈련용 시뮬레이터 개발", 한국전력공사. (1990)
9. "CANDU형 발전시스템 설계·운전 지원용 플랜트-제어 모사 해석기 구현", KAERI/CM-149/94
10. "DCC Emulator for CANDU", AECL.

그림 1. 하드웨어 구성도

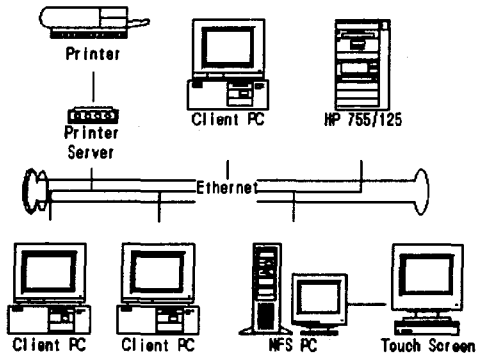


그림 4. 사용자 인터페이스 구성도

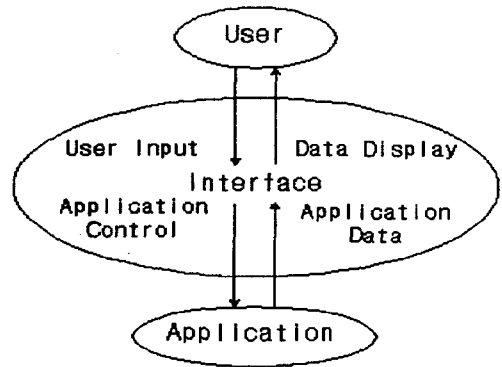


그림 2. 그래픽 개발 물의 Block 구성도

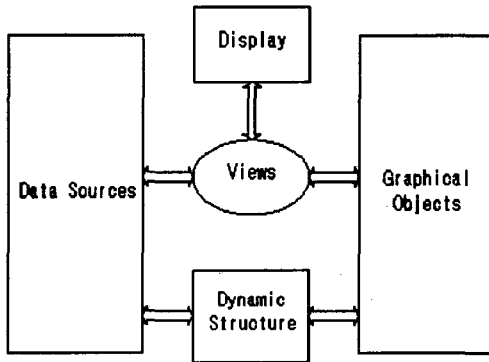


그림 5. 인터페이스 프로그램의 흐름도

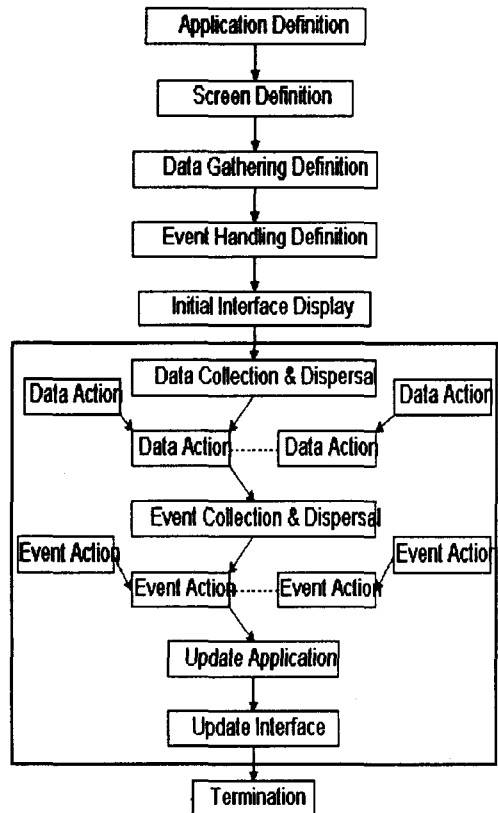


그림 3. 시뮬레이터 디스플레이 시스템의 View

