

발전소 시뮬레이션을 위한 사용자 인터페이스 설계 및 구현

피무호^o 최종필
한국산업기술대학교 컴퓨터공학과
{pi79^o, jchoi}@kpu.ac.kr

A Design and Implementation of User Interface for Power System Simulation

Mu-ho Pi^o Jong-pil Choi
Dept. of Computer Engineering, Korea Polytechnic University

요 약

발전소 시뮬레이터는 발전소 설계의 검증, 제어 시스템의 성능 시험, 제어 시스템 상수 최적화, 새로운 제어 기법의 사전 검증 및 운전요원 훈련과 같은 다양한 목적으로 사용되어진다. 본 논문에서는 발전소 운전요원들의 운영 능력 향상을 위하여 사용자의 이벤트를 저장하여 사용자 요구 시에 일정 시간 주기로 도면에 설계하는 스크립트 기능과 그래픽 기반의 시뮬레이션을 위한 두 종류의 결과 모니터링 윈도우 그리고 사용자의 다양한 시뮬레이션을 위한 시뮬레이션 속도 변경 기능을 기존의 발전소 시뮬레이터에 설계, 구현한다. 또한 시뮬레이션 속도 변경 기능 확장에 따른 시뮬레이션의 정확성을 판단하기 위하여 수행 결과 데이터를 추출하여 시뮬레이션의 정확성을 검증한다.

1. 서 론

발전소 시뮬레이터는 발전소 설계의 검증, 제어 시스템의 성능 시험, 제어 시스템 상수 최적화, 새로운 제어 기법의 사전 검증, 동특성 해석, 운전 절차서 개발 및 검증 그리고 운전요원 훈련과 같은 다양한 목적으로 사용될 수 있다.

특히 원자력 발전소의 안전성과 신뢰성확보를 위해서는 발전소를 운전, 유지하는 운전요원의 대처능력향상이 무엇보다도 필요하고 운전요원 훈련을 통해 발전소 운전의 안전성과 신뢰성을 확보하거나, 유지하기 위해 시뮬레이터를 사용하는 것은 현재 세계적으로 일반화되어있다. 특히 1979년 TMI(Three Mile Island) 원자력 발전소 사고 분석 결과, 주제어실(Main Control Room)에서 발전소 운영을 맡고 있는 운전요원의 조작 미숙과 대처부족이 주요 원인으로 판명되어 시뮬레이터의 필요성은 더욱 대두되었다[1].

원자력 발전소의 안전성 확보를 위해 가장 강조되었던 대책중의 하나는 효과적인 사고대처 능력 배양을 위해 전 범위(Full Scope) 시뮬레이터를 이용한 운전요원 훈련이다 [4].

기술 발달에 의해 발전 설비가 고기능화 되어 발전소의 정지 횟수가 줄어들고 운전요원이 운전조작을 할 수 있는 기회가 감소함에 따라 발전소 비정상 상태 발생 시에 긴급 대처 능력향상을 위하여 운전요원 훈련이 절실히 요구되었다.

본 논문에서는 운전요원의 조작 능력을 향상시키기 위하여 기존의 발전소 시뮬레이터[2]에 확장된 형태의 GUI 시뮬레이션 기능을 구현한다.

확장된 기능으로는 사용자의 각 이벤트를 저장하여 사용자 요구 시에 일정시간 주기로 도면에 설계를 하는 스크립트 기능과 그래픽 기반의 시뮬레이션을 위하여 선형 그래프와 계측기 방식의 결과 모니터링 윈도우, 사용자의 다양한 시뮬레이션을 위한 시뮬레이션 속도 변경 기능이다. 또한 시뮬레이션 속도 변경 기능 확장에 따른 데이터의 정확성을 판별하기 위하여 수행 결과 데이터를 추출하여 시뮬레이션의 정확성을 검증한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구로 기존 발전소 시뮬레이터를 비교하며 3절에서는 운전 요원 조작 능력의 향상을 위한 기능들의 구현 방법을 언급하며 속도 변경 시뮬레이션 수행 결과의 정확성을 검증한다. 4절에서는 결론을 맺는다.

2. 관련 연구

본 논문에서 소개할 운전요원의 조작 능력 향상을 위한 확장된 형태의 발전계통 시뮬레이터의 구현에 앞서 기존에 사용되어지고 있는 국외의 상용화 발전소 시뮬레이터들의 특징을 소개한다.

국외의 발전소 시뮬레이터 틀로는 TRAX가 있다. TRAX는 엔지니어링 등급의 소프트웨어로서 GUI 기반의 시뮬

레이션 기능을 제공하고 있으며 상용화 프로그램이다. 현재 열병합 발전소 운전자를 위하여 100여개의 계통 시뮬레이터(보일러, 가스, 제어기기 등)를 제공하고 있지만 각각의 계통에 해당되는 시뮬레이터 툴이 개별적으로 존재한다. 또한 툴의 구성이 사용자가 아닌 개발자의 편의에 맞게 구성 되어있어서 사용방법이 어려우며 결과 모니터링 방법이 단조로운 단점을 가지고 있다.

국내의 GSim은 일부 대학에서 교육용으로 사용되어지고 있는 소프트웨어로서 GUI 기반의 시뮬레이션 기능을 제공하고 있으며 오픈 소스 프로그램이다. 현재 발전계통 시뮬레이션을 위하여 중기, 급수, 단일 계통의 시뮬레이터를 제공하고 있지만 시뮬레이션 기능이 미흡하다.

3. 구현

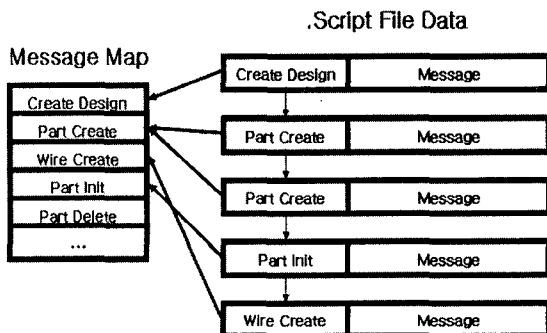
3.1 스크립트 기능

스크립트 기능은 도면 설계 작업을 이벤트 단위로 분할하여 파일로 저장하고 사용자 필요시에 일정 시간 주기로 도면에 표현하는 기능으로 사용자가 수행하였던 전체 이벤트를 순차적으로 확인하며 검증할 수 있도록 한다.

<표 1> 스크립트 메시지 맵

CREATE_DESIGN	디자인 파일 생성
PART_CREATE	컴포넌트 생성
WIRE_CREATE	링크(플로우, 콘트롤) 생성
PART_INIT	컴포넌트 파라미터 입력
PART_DELETE	컴포넌트 삭제
WIRE_DELETE	링크(플로우, 콘트롤) 삭제
PART_MOVE	컴포넌트 이동
WIRE_MOVE	링크(플로우, 콘트롤) 이동

스크립트 시작을 수행하면 [표 1]의 스크립트 메시지 맵과 일치하는 사용자 이벤트가 발생할 때, 이벤트들을 메모리에 저장하며 스크립트를 종료하게 되면 메모리에 저장되어있는 이벤트 목록들을 파일로 저장한다. 저장 방식은 STL (Standard Template Library)[3]의 리스트 구조를 사용하였다.



[그림 2] 메시지 맵과 파일 데이터의 매핑

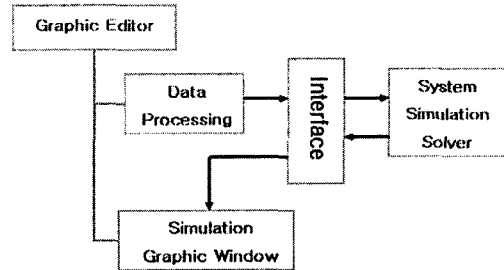
저장된 스크립트 데이터와 메시지 맵 간의 매핑은 [그림

2]와 같다. 사용자가 스크립트 수행을 요청하게 되면 파일에 저장되어있는 스크립트 데이터를 읽어 들여 메시지 맵과 일치하는 이벤트를 일정시간 주기로 도면에 보여주게 된다.

3.2 결과 모니터링

결과 모니터링은 시스템의 운영 시에 각 서브 시스템이나 모듈의 운영 상태를 다양하고 친숙한 형태로 시뮬레이션하는 방식을 뜻하며 이 기능을 통하여 시스템의 운영 상황을 파악할 수 있다.

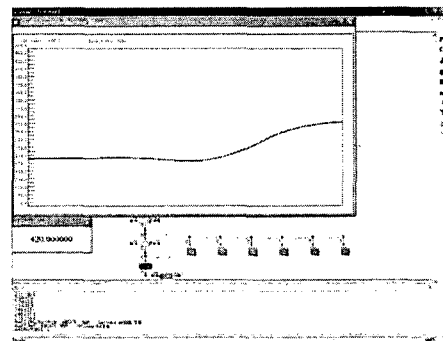
결과 모니터링 방식은 두 가지 형태로 지원된다. 첫 번째 방식은 선형 그래프 방식으로 시뮬레이션 결과를 그래프로 표현한다. 두 번째 방식은 계측기 형태로 실제 발전소 운영원의 환경에 친숙한 형태로 결과를 보여준다.



[그림 3] 결과 모니터링 윈도우와 시뮬레이터 연결 구조

[그림 3]은 그래픽 에디터에 속해 있는 결과 모니터링 윈도우와 시스템의 시뮬레이터에 해당하는 시뮬레이션 solver와의 연결 구조를 보여주고 있다. 사용자가 시뮬레이션을 수행하게 되면 solver에 데이터 처리 요청을 보내며 solver는 데이터 처리 요청을 받게 되면 시뮬레이션을 수행하게 되며 1cycle의 수행이 완료되게 되면 결과 값을 결과 모니터링 윈도우에 보내주게 된다.

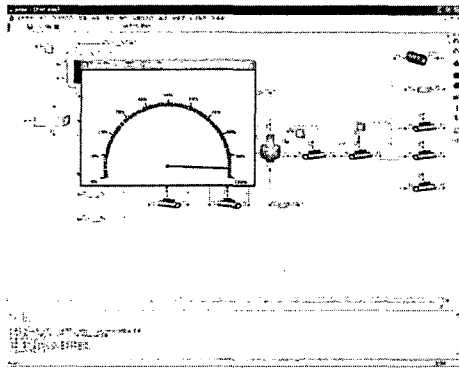
선형 그래프 방식은 FIFO 알고리즘을 사용하여 solver에서 받은 데이터들을 버퍼에 저장하며 모니터링 윈도우에 결과를 표현한다. 버퍼의 크기는 모니터링 윈도우의 x좌표 시작과 끝에 해당되며 버퍼가 전부 채워지면 가장 마지막에 입력된 데이터를 삭제하고 새로운 데이터를 버퍼의 마지막에 채우게 된다. 이 버퍼의 데이터들을 선형으로 연결하여 결과 모니터링을 보여주게 된다.



[그림 4] 선형 그래프 방식의 결과 모니터링

[그림 4]는 FIFO 알고리즘을 이용하여 수행되는 선형 그래프 결과 모니터링 윈도우의 실행 결과 화면을 보여주고 있다.

계측기 방식은 삼각함수를 통한 원 그리기 알고리즘을 이용하여 수행된다. 수행될 결과의 최대값과 최소값을 백분율로 반원의 시작과 끝점으로 하며 solver에서 받은 데이터에 해당되는 지점을 기점으로 하여 원의 중점에서 연결하는 방식을 사용하여 화면상에 보여지게 된다. 수행 결과 화면은 [그림 5]와 같다.



[그림 5] 계측기 방식의 결과 모니터링

3.3 수행 속도

시뮬레이션을 위한 계산 수행 방식은 연속시간과 이산시간 시뮬레이션 방식의 혼합형으로 어느 한 순간에 각 모델 컴포넌트들의 파라미터 속성 값을 결정하기 위한 계산은 연속시간 시뮬레이션 방식과 같은 연속 계산 모델에 의해서 계산이 이루어지며, 계산이 종료됨과 동시에 각 컴포넌트의 파라미터 속성 값이 새로운 값으로 변경되어진다. 이러한 연속 계산을 일정 시간마다 반복하게 되는데 사용자가 계산 속도를 변경하여 수행할 수 있도록 지원한다. 수행 시간에 따라 수행되는 결과의 검증 표는 <표 2>와 같다.

<표 2> 수행 시간 변경에 따른 실행 결과 검증

시간 (sec)	1배	4배	8배
0.25	175.999201	175.943750	175.943750
0.50	176.055641	176.171498	176.415188
0.75	176.113073	176.415188	176.951045
1.00	176.171498	176.674981	177.552852
1.25	176.230920	176.951045	178.223220
1.50	176.291341	177.243574	178.968645
1.75	176.352763	177.552852	179.866873
2.00	176.415188	177.879219	180.589316
2.25	176.478621	178.223220	181.364451
2.50	176.543062	178.586288	182.193926
2.75	176.608514	178.968645	183.080882
3.00	176.674981	179.370182	184.031040
3.25	176.742465	179.866873	185.053581
3.50	176.810968	180.221577	186.219426
3.75	176.880495	180.589316	187.167517

4.25	177.022625	181.364451	189.194877
4.50	177.095242	181.772244	190.279525
4.75	177.168886	182.193926	191.414111
5.00	177.243574	182.629936	192.600583
5.25	177.319313	183.080882	193.840756
5.50	177.396097	183.547486	195.136235

[표 2]의 수행 결과 검증 표와 같이 수행 속도 변경에 따른 각 수행 결과는 속도 변화와 일치하고 있다. 8배 속도의 수행 결과는 1배 속도 수행의 8의 배수에 해당하는 2초 수행시의 결과 값을 첫 수행 이후에 수행하게 되며 4배 속도의 수행 결과는 1배 속도 수행의 4의 배수에 해당하는 1초 수행시의 결과 값을 첫 수행 시간 이후에 수행하게 된다. 이와 같은 수행 검증을 통하여 사용자가 수행 속도를 변경해도 같은 수행 결과가 나온다는 것을 검증하였다.

4. 결론 및 향후 과제

본 논문에서는 기존의 발전소 시뮬레이터 틀에 운전요원의 조작 능력 향상을 위한 사용자 인터페이스를 확장하였다. 스크립트 기능을 통하여 사용자는 자신의 수행 작업들을 순차적으로 파악하며 디버깅할 수 있으며 그래프를 통한 수행 결과 화면을 통하여 보다 친숙한 결과 시뮬레이션을 할 수 있다. 또한 수행 속도 변경에 따른 실행 결과를 검증하여 사용자가 수행속도를 변경하더라도 시뮬레이션이 가능하도록 설계, 구현하였다. 향후 과제로서 현재 네트 연결 상태를 점검하여 사용자에게 연결 오류 정보를 알리는 기능과 추가적인 사용자 인터페이스를 설계할 예정이다.

참 고 문 헌

- [1] G.E Cummings, "Operator/Instrumentation Interaction During The Three Mile Accident", IEEE transaction on Nuclear Science, vol. NS-27, 1980.
- [2] 피무호, 최종필, "GUI 기반의 발전계통 시뮬레이터의 설계 및 구현", 한국정보과학회 추계학술대회, 2004.
- [3] Josuttis, "The C++ Standard Library : A Tutorial and Reference", Addison-Wesley, 1999.
- [4] 박근욱, 이종복, 서용석, 구인수, "설계해석코드 기반의 원자력발전소 훈련용 시뮬레이터 개발전략", 대한산업공학회/한국경영과학회 춘계공동학술대회, 2000.