

다분야통합최적설계를 위한 설계프레임워크: EMDIOS (Extensible Multidisciplinary Design Integration and Optimization System)

이 세 정*

(서울시립대학교 기계정보공학과)

제품설계에 관련된 다양한 공학해석분야, 즉 구조해석, 동역학, 열·유체·유동해석, 제어, 전자기장해석 등을 동시에 고려하면서 최적의 설계를 결정하는 것을 다분야통합최적설계(Multidisciplinary Design Optimization: MDO) 기술이라 한다. MDO 프레임워크(framework)는 최적화기술, 컴퓨팅기반구조기술, 통합설계기술이 유기적으로 구현된 소프트웨어 복합체계로서, 분산컴퓨팅 기반구조를 통하여 MDO 요소 기술들과 기존의 CAD/CAE 도구들을 연계하여 설계 작업을 통합적으로 관리하고 자동화한다. MDO 프레임워크는 이러한 자동화된 통합관리를 통하여 설계도구 간의 데이터 전달과 변환에 소요되는 설계자의 부담을 경감시키며 다분야 전문가가 참여하는 공동 작업 환경을 제공함으로써 설계 효율성을 증진시킨다. 이 글에서는 현재 최적설계신기술 연구센터(iDOT)에서 개발 중인 MDO 프레임워크인 EMDIOS를 소개하기로 한다.

1. 개발 배경

현재 산업체의 요구를 살펴 보면, 다분야 설계문제를 해결하기 위한 도구의 필요성은 크게 인식되고 있다. 또한 공학설계를 지원하는 통합 소프트웨어 프레임워크의 상품적 가치가 점차 부각되고 있으며 이에 따른 관련 상품이 다양하게 출시되고 있는 상황이다. 하지만, 기존 시스템들은 현장 설계자들이 사용하기에 기능이 충분하지 않으며, 다분야통합설계방법론의 적용에 있어서도 매우 미흡한 수준이라 판단된다.

다분야통합최적설계 기술은 다양한 분야의 공학적 설계 원리들을 동시에 고려하여, 균형 있고 유기적인 방법으로 최적의 설계를 결정하는, 체계적인 설계자동화 기술로서 짧은 시간에 적은 비용으로 경쟁력 있는 제품의 설계를 가능하게 할 것으로 기대되는 기술이다. 다양한 수요자의 요구를 만족하는 복잡한 공학 시스템에 대한 수요의 증가로 이러한 MDO 기술 및 프레임워크의 개발 필요성이 점차 증대되어 왔다.

일반적으로 프레임워크이란 실제 응용프로그램의

응도에 맞는 주문제작(customization)이 가능한 일종의 전단계 프로그램을 말한다. 따라서 MDO 프레임워크는 MDO 기술이 구현되어 재사용이 가능한 반 완성(semi-completed) 형태의 프로그램이며, MDO 방법론을 MDO 프레임워크 상에 구현하여 빠르고 정확하게 최적설계 작업을 수행할 수 있게 된다.

2. 요구사항 분석

프레임워크를 개발하기 위해서는 어떤 작업을, 어떠한 방식으로 처리하고자 하는지에 대한 요구사항 분석이 있어야 한다. 아래는 실제 설계를 하는 사용자와 최적설계 방법론 개발자, 기반구조 개발자들이 협의하여 결정한 다섯 가지 요구조건들이다.

2.1. 분산 환경

MDO 문제를 해결하기 위해 사용되는 여러 도구들은 네트워크로 연결된 서로 다른 컴퓨터 상에서 작동하고 있는 경우가 많다. 분산 환경에서 작동되는 각 도구들은 몇 초에서 몇 시간 또는 며칠에 이르기까지 그 수행 시간이 다양하다. 따라서, 이러한 도구들은 가능하다면, 분산 환경에서 병렬적으로 수행될 때 높은 성능을 얻을 수 있다. 이러한 필요

*E-mail: sclee@uos.ac.kr

성 때문에 프레임웍은 분산 환경을 지원해야 한다.

프레임웍에서 필요로 하는 분산 환경의 요구 사항은 CORBA 나 EJB에서 제공하는 복잡한 서비스를 필요로 하지는 않고, 단지 각 사이트에 떨어져 있는 도구들이 서로 데이터를 교환할 수 있는 기능이 필요하므로, 원격 프로시저 호출이 가능하면 된다.

2.2. 데이터 관리

MDO 문제를 해결하기 위해서는 다양한 도구들이 사용되며, 하나의 도구에서 발생하는 출력은 다른 도구의 입력으로 사용될 수 있다. 산업 현장에서 사용되는 대부분의 도구들은 그들 나름대로 정의된 형식으로 파일에 데이터를 저장한다. 도구마다 저장하는 데이터 형식이 다르기 때문에, 하나의 도구에서 발생하는 출력을 다른 도구의 입력으로 사용하기는 쉽지 않다. 만약 자동화된 통합 환경이 제공되지 않는다면, 두 도구를 모두 잘 알고 있는 전문가가 입력 파일과 출력 파일의 내용을 비교해서, 연관 있는 데이터를 추출하여 사용하여야 한다.

산업 현장에서의 MDO 문제를 해결하기 위해 사용되는 도구들은 그 수행 시간이 길어서 며칠을 소요하는 경우도 있을 수 있다. 따라서, MDO 문제 풀이를 일시 중지한 후 이를 재 시작할 수 있어야 하는데, 이를 위해서는 각 도구들이 사용했던 작업 데이터들을 저장했다가 다시 사용할 수 있도록 지원해야 하며, MDO 문제 풀이의 중간 결과도 유지하고 있어야 한다. 따라서, MDO 프레임웍은 다분야 전문가들이 공통의 데이터를 안전하게 공유할 수 있도록 지원해야 한다.

2.3. 사용자의 편의성

MDO 프레임웍을 사용하는 사용자의 수준은 다양하다. MDO 방법론을 이미 숙지하여, 직접 MDO 방법론을 적용하여 MDO 문제를 설정하고, 도구들을 연결하여 사용할 수 있는 고급 사용자에서부터, MDO 방법론에 대해서 자세히 알지 못하지만 이미 해결해 놓은 유사한 문제를 재활용해서 문제를 해결하려는 초보사용자에 이르기까지, 서로 다른 수준의 사용자들을 위한 계층적인 시각이 제공되는 것이 바람직하다.

고급 사용자는 자신이 필요로 하는 새로운 도구

를 MDO 프레임웍에 쉽게 통합할 수 있기를 바랄 것이고, 복잡한 문제 설정이 가능하기를 원할 것이다. GUI(Graphical User Interface)를 이용하면 간단한 문제를 설정하는 데는 편리할 수 있지만, 반복적이며 단순한 작업을 너무 많이 해야 하는 경우도 있을 수 있다. 또한, 복잡한 문제를 설정하기 위해서는 다양한 제어가 가능해야 하므로, 이를 GUI로는 모두 지원하지 못하는 경우도 있을 수 있으며, 설정 설정할 수는 있을지언정 너무 복잡하여 관독하지 못하는 경우도 있을 수 있다. 따라서, GUI와 더불어 복잡한 문제를 설정할 수 있는 스크립트 언어가 제공되어야 한다.

스크립트 언어는 기능적으로 GUI의 수퍼셋(superset)이어야 하며, 스크립트 또는 GUI를 통해 설정된 문제에 대한 정의 및 문제 풀이 과정에서 생성된 데이터, 결과 데이터 등은 데이터베이스에 저장되어 있어서 언제든지 확인 또는 재사용 가능해야 한다.

초보 사용자들은 복잡한 조작 없이 문제 풀이가 가능하기를 원하므로, 편리한 GUI를 기본적으로 제공해야 할 것이며, 반드시 입력이 필요한 데이터가 아니라면 디폴트(default) 값으로 작동할 수 있도록 지원해야 한다. 또한, 이미 풀어 놓은 유사한 문제를 이용하여 문제를 해결할 수 있도록 지원해야 할 것이며, 템플릿(template) 기능을 제공하는 것이 바람직하다. 이의 지원을 위해서는 이미 해결해 놓은 문제를 관리하는 것이 필요하다.

2.4. 응용 프로그램 인터페이스

MDO 프레임웍은 각 공학해석 분야의 원리들이 구현된 도구들이 통합되어 작동할 수 있도록 지원해야 한다. 하지만, 각 도구들은 통합을 고려하지 않고 독자적으로 개발되었기 때문에, MDO 프레임웍 상에서 다른 도구들과 같이 통합적으로 작동하기 위해서는 독자적으로 개발된 도구들에게 래퍼(wrapper)를 씌우는 방법을 사용할 수 있다.

각 도구들이 분산 환경에 흩어져 있을 수 있으므로, MDO 문제를 제대로 설정하기 위해서는 사용자에게 각 도구들에 대한 메타 데이터를 제공할 필요가 있다. 즉, MDO 프레임웍은, 어떤 도구가 어떤 사이트에 존재하는지, 해당 도구를 사용하기 위해 필요한 파일 또는 기본 옵션이 무엇인지에 대한

정보를 제공해야 한다.

MDO 프레임워크는 각 공학해석 분야의 원리들이 구현된 도구들이 통합되어 작동할 수 있도록 지원해야 한다. 이는 새로운 도구가 개발되었을 경우, 새로운 도구 또한 기존의 MDO 프레임워크에 쉽게 통합되어 작동될 수 있어야 함을 의미한다. 즉, MDO 프레임워크는 확장 가능해야 하며, 이렇게 새로이 추가되는 도구들에 대한 정보도 시스템에서 관리할 수 있어야 한다.

2.5. 설계 도구 통합

MDO 문제에서는 각 분야에 따라 다양한 종류의 CAD/CAE 도구, 최적화 코드가 사용되므로, 프레임워크는 이들 도구들이 통합되어 작동될 수 있는 환경을 제공하는 것이 필수적이다. 위에서 언급한 요구 사항들이 모두 만족될 때 설계 도구 통합이 이루어진다고 할 수 있다.

3. EMDIOS의 개발

개발하는 프레임워크는 분산 환경에서 작동하기 때문에 현재 가용한 도구들이 어떤 것이 있는지에 대한 정보가 제공되어야 하며, 멀리 떨어져 있는 컴퓨터에 있는 도구를 작동시킬 수 있는 방법이 있어야 한다. 우선 제안하는 MDO 프레임워크에서 사용하는 용어를 소개한다. 제안하는 MDO 프레임워크에서 통합되어 사용될 수 있는 도구를 컴포넌트(component)라 지칭한다. 컴포넌트는 그 상태에 따라, 후보 컴포넌트(candidate component), 수행가능한 컴포넌트(runnable component), 수행중인 컴포넌트(running component)로 나뉜다.

새로운 도구를 MDO 프레임워크에 통합하여 사용하려고 하면 해당 도구를 MDO 프레임워크에 등록해야 한다. 예를 들어, ANSYS와 같은 CAE 프로그램을 MDO 프레임워크에 통합하여 사용하려고 한다면, ANSYS를 등록해야 한다. 통합하여 사용하려고 하는 프로그램의 이름, 종류, 수행하기 위해 필요한 명령어, 해당 프로그램이 어느 컴퓨터에 있으며(해당 컴퓨터의 URI) 어떤 디렉토리에 위치하는지에 대한 정보가 MDO 프레임워크에 등록된다. 이런 절차를 통해 등록된 도구를 후보 컴포넌트(candidate component)라고 부른다.

후보 컴포넌트가 최적설계문제를 위해 사용되기 위해서는 특정 문제에 맞게 입력 파일과 출력 파일의 내용이 정의되어야 한다. 또한, 해당 문제에서 문제를 해결하기 위해 입력 파일의 어떤 부분의 값을 변경해 가면서 시뮬레이션을 수행하여 최적의 값을 얻어낼 지를 결정해야 한다. 또한, 출력 파일 내에 어떤 부분이 최적화하고자 하는 값인지를 지정해야 한다. 이러한 것이 지정된 후보 컴포넌트를 수행가능한 컴포넌트(runnable component)라 한다.

후보 컴포넌트를 수행가능한 컴포넌트로 만드는 과정은 래퍼(wrapper)를 통해 진행된다. 여러가지 래퍼중에서 가장 많이 사용되는 파일 래퍼는 래퍼 GUI를 통해서 입출력 파일의 필요 부분을 지정한다. 사용자가 지정한 내용은 스크립트 형태로 저장되며, 추후 해당 컴포넌트가 최적화 문제 풀이를 위해 사용될 때, 이 스크립트 내용이 파일 래퍼에 의해 처리된다.

수행가능한 컴포넌트가 문제 풀이를 위해 실제 작동되면 수행중인 컴포넌트(running component)가 된다. 파일 래퍼는 해당 도구가 사용될 때 마다 사용자에게 의해 생성된 스크립트에 따라, 수행중인 컴포넌트가 이번 시뮬레이션에서 필요한 입력 데이터를 데이터 서버에서 가져와서 입력 파일의 필요 부분에 삽입한다. 또한 컴포넌트의 시뮬레이션 작동이 끝나면, 스크립트에 정의된 내용에 따라 출력 파일의 지정된 위치에서 필요한 출력 데이터를 추출하여 데이터 서버를 통해 데이터베이스에 저장한다. MDO 프레임워크는 분산 환경에서 작동하므로, 각 사이트마다 해당 사이트에서 작동할 컴포넌트를 시작시키고, 정시키는 데몬(demon) 프로그램인 CSA(Component Service Agent)가 작동한다.

데이터서버는 크게 두 가지 종류의 데이터를 저장한다. 하나는 메타 데이터(meta data)이고, 다른 하나는 실제 데이터(instance data)이다. 메타 데이터는 위에서 설명한 컴포넌트에 대한 정보, 정의된 최적화 문제에 대한 정보를 포함한다. 실제 데이터는 최적화 문제 풀이 과정에서 발생한 데이터들을 포함한다.

프레임워크에서는 메타 데이터를 위한 XML 파일의 구조를 정의하는 스카마 파일들을 정의하여 정의된 메타 데이터가 유효한 형태를 가지고 있는지 검사한다.

데이터 서버는 기본적으로 데이터베이스를 사용하고 있으며, XML 형태의 정보를 저장하는 방법에는 여러 가지 방법이 있을 수 있다. 본 프레임워크에서 메터 데이터는 XML 파일 형태로 정의하여, 테이블의 한 컬럼(column)에 그대로 저장하였다. 따라서, 메터 데이터의 저장 단위는 XML 파일이고, 컴포넌트간의 메터 데이터의 정보 전달의 단위도 XML 파일이다. 이와는 달리, 문제 풀이 과정에서 발생하는 데이터는 테이블 형태로 저장된다.

4. 구성 요소

여기서는 제안하는 MDO 프레임워크를 구성하는 각 구성요소 프로그램들을 살펴본다.

4.1. 사용자 프로세스(User process)

사용자가 MDO 프레임워크와 대화하기 위해 사용하는 GUI이다. 새로운 도구를 MDO 프레임워크에 통합할 수 있는 GUI를 제공하며, 현재 사용 가능한 도구들에 대한 정보를 열람할 수 있으며, 도구들을 이용하여 문제를 설정하고, 설정된 문제를 직접 실행시킬 수 있다

4.2. 스케줄러(Scheduler)

스케줄러는 사용자가 정의한 문제 풀이 순서에 따라 실제로 문제를 해결하는 과정을 관리한다. 사용자가 정의한 MDO 문제에 대한 정의는 내부적으로 데이터 서버를 통해 저장되며, 문제 풀이 순서는 스크립트 파일로 저장된다. 사용자가 GUI를 통해 문제 풀이 순서를 정의할 수도 있고, 스크립트를 통해 직접 정의할 수도 있는데, GUI를 통해 문제 풀이를 구성하는 경우, 그와 대등한 스크립트가 생성되어 관리된다. 스케줄러는 정의된 스크립트 파일을 기초로 문제 풀이를 실행한다. 스케줄러는 독자적인 프로세스로 작동하며, 데이터 서버 또는 다른 구성 요소들과 통신하며 문제를 해결한다.

4.3. 데이터 서버(Data Server)

데이터 서버는 데이터베이스와 연계되어 분산된 사이트에 대한 정보, 각 도구들에 대한 정보, 문제 정의에 대한 정보, 문제 풀이시 발생하는 데이터들을 체계적으로 관리한다. 데이터 서버는 제공되는

데이터베이스 관리 시스템 위에서 작동하는 독립적인 프로세스로 구성된다. MDO 프레임워크의 각 구성 요소들은 항상 데이터 서버를 통해서 데이터를 공유하며, 구성 요소끼리 직접적으로 데이터를 주고받지는 않는다.

4.4. 도구들을 위한 래퍼

래퍼는 데이터 서버를 통해 도구들의 작동에 필요한 데이터를 추출하고, 도구들이 생산한 데이터를 데이터 서버에 전달하는 역할을 수행한다.

독자적으로 개발된 도구는 크게 두 가지 형태로 나누어 볼수 있는데, 하나는 독립적인 프로세스로 작동하는 경우이며, 다른 하나는 함수 형태로 제공되는 경우이다.

많은 CAE들은 독립적인 프로세스로 작동한다. 대부분의 CAE들은 입력 파일을 받아 들여 처리한 후 출력 파일을 생성한다. 앞에서 언급한 파일래퍼가 다른 구성 요소에서 생성한 데이터를 해당 CAE를 위해 입력 파일에 반영해 주며, 시뮬레이션이 끝난 후 출력 파일에서 필요한 데이터를 추출해서 데이터 서버에 전달해 준다.

MDO 문제 풀이에 사용되는 도구들은 위에서 언급한 CAE, 최적화기 외에 CAD 도구가 있다. CAD 도구는 분해 풀이 과정 내에서 반복적으로 호출되기 보다는 분해 설정 단계 또는 분해 해결 후 결과를 출력하는 용도로 사용될 수 있다. 따라서, 문제 풀이 과정에서 반복적으로 호출되는 CAE나 최적화기와는 다른 특성을 가지며, 문제 풀이 과정에서 CAD와의 인터페이스는 중요하지 않다. 해결된 문제의 결과를 CAD GUI로 출력할 수 있도록 하는 것은 사용자 편의상 제공될 필요가 있지만, 이번 글에서는 이 부분은 다루지 않는다.

4.5. CSA(Component Server Agent)

분산 환경에서는 여러 사이트에 도구들이 흩어져 있을 수 있기 때문에, 도구들이 작동해야 하는 사이트마다 데몬 프로세스가 작동할 필요가 있다. CSA은 새로운 도구를 등록할 때 해당 도구가 해당 사이트에 실제로 존재하는지를 확인해 줄 수 있으며, 실제로 도구가 분해 풀이에 사용되는 경우, 해당 도구의 래퍼를 작동하게 하여 문제 풀이에 사용될 수 있도록 한다.

이러한 도구들을 이용하여 문제를 해결하여 나가는 과정을 개괄적으로 살펴보면 다음과 같다.

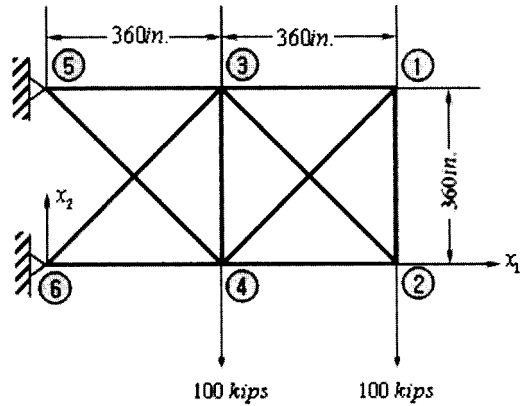
사용자가 MDO 프레임워크에서 제공되는 사용자 GUI를 통해 MDO 문제를 설정하면, 문제에 대한 정의가 데이터 서버를 통해 데이터베이스에 저장되며, 해당 문제를 위해 설정한 문제 풀이 순서는 스크립트 파일로 저장된다. 이때 사용자는 현재 사용 가능한 도구들과 이들 도구들이 존재하는 사이트에 대한 정보를 GUI 상에서 데이터 서버를 통해 추출해 낼 수 있으며, 이들 중 어떤 도구들을 사용할지를 결정할 수 있다. 사용자가 자신이 정의한 문제 풀이 순서에 따라 문제 풀이를 작동시키면, 스케줄러가 스크립트 파일에 저장된 내용을 기반으로 문제 풀이를 시작하게 한다. 이때 해당 문제 풀이 과정에서 사용되도록 지정된 도구들이 작동될 준비가 되어 있는지를 각 사이트에 존재하는 CSA를 통해 확인하며, 만약 준비되어 있지 않다면 CSA가 이들 도구들이 작동될 수 있도록 준비시킨다. 준비가 완료되면, 문제 풀이가 진행된다. 문제 풀이가 진행되는 동안 문제 풀이 순서에 정의된 순서에 의해 각 도구들이 작동하며, 래퍼들이 해당 도구를 위해 필요한 데이터를 데이터 서버에서 추출하고, 해당 도구가 생산한 데이터를 데이터 서버에 저장한다. 스크립트 파일에 정의되어 있는 문제 풀이 종료 조건이 만족될 때까지 문제 풀이는 반복적으로 진행된다. 문제 풀이 종료 조건이 만족되면 사용자는 최적화된 변수 값을 받아볼 수 있다.

5. 최적화 예제 실행

현재까지 개발된 EMDIOS의 기능 및 성능을 검증하기 위하여 여러 가지 최적화 문제들을 실행하였는데, 여기서는 문헌에서 빈번히 사용되는 예제인 10-Bar truss 중량 최소화를 EMDIOS에서 실행하는 과정을 소개한다.

5.1. 문제 정의

10-bar truss 문제는 그림 1과 같이 10개의 봉으로 구성된 트러스 구조물에서 좌측단 5번과 6번 절점이 고정되어 있고, 2번과 4번 절점에 100 kips의 하중이 작용할 때, 각 부재에 작용하는 응력과 절점의 처짐량이 각각 허용응력(± 25.0 ksi)과



$$F_{obj} = \sum_{i=1}^{10} \rho_i L b_i$$

$$g_i = |\sigma_i| - \sigma^o \leq 0 \quad i = 1, 2, \dots, 10$$

$$g_{10+j} = |z_j| - z^o \leq 0 \quad j = 1, 2, \dots, 6$$

Modulus of elasticity = 10^4 ksi

Material density = $0.101b/in^3$

Lower limit on area = $0.10in^2$

Displacement limit = $2.0in$

Stress limit = $\pm 25.0ksi$

그림 1. 10-bar truss

허용범위(2.0 in)를 만족하면서 구조물의 전체중량을 최소화하는 각 봉의 단면적을 구하는 구조최적설계문제이다. 여기서 봉의 단면적이 0이 되는 것을 방지하기 위하여 하한값($0.10in^2$)을 설정하였다.

5.2. 시스템 구성

10-Bar truss의 각 단면적을 입력으로 받아, 응

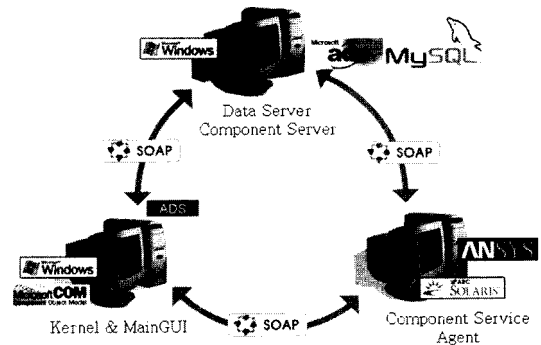


그림 2. System Architecture

력 및 변위를 계산해 주는 프로그램은 상용 구조 해석 프로그램인 ANSYS/Multiphysics 5.6을 사용하였다. 그리고, 본 프레임워크 설계의 핵심 요소 중 하나인 분산 환경 지원을 테스트 하기 위하여, Solaris OS를 탑재한 SUN 워크스테이션에 ANSYS 5.6 및 CSA를 설치하고, File Wrappger GUI 에 데이터를 이용하여 수행가능한 컴포넌트를 만들었다. 전체적인 시스템 구성은 그림 2와 같다.

5.3. 최적화 프로세스 정의

MDO 프레임워크에서 제공하는 시각 프로그래밍 툴 과 스크립트 언어를 사용하여, 최적화 프로세스를 정의한다. 그림 3에서 "Bar10"으로 표현된 블록이 ANSYS와 파일 랩퍼 GUI를 사용하여 만든 수

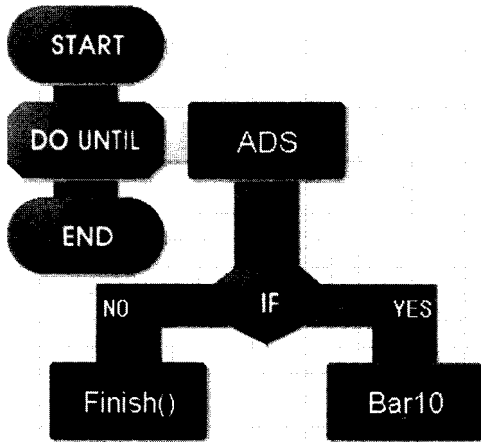
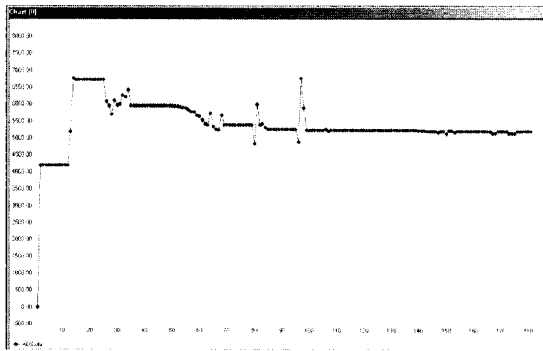


그림 3. Control flow



(a) Chart

행가능한 컴포넌트이며, "ADS"는 ADS 최적화 모듈을 수행가능한 컴포넌트로 만든 것이다(그림 3).

5.4. 모니터링

위에서 정의된 최적화 프로세스를 실행하면, 그림 4와 같이 그 결과가 동적으로 사용자가 정의한 출력 형식(테이블 또는 차트)로 화면에 표시된다.

6. 요약

MDO 프레임워크는 설계 작업을 통합적으로 관리하고 자동화하여 설계도구 간의 데이터 전달과 변환에 소요되는 설계자의 부담을 경감시키며 다분야 전문가가 참여하는 협동설계 환경을 제공함으로써 다분야를 동시에 고려한 효율적 설계를 지원한다.

이번 글에서는 이러한 MDO 프레임워크로 개발된 EMDIOS를 소개하고 그 개발 배경과 타당성을 개략적으로 제시하였다. EMDIOS는 분산 환경을 제공하고, 데이터베이스와 연계되어 정의된 문제와 문제 풀이 절차를 저장하고 실행 시 발생하는 데이터를 체계적으로 관리하는 구조를 갖는다.

MDO 문제 해결에 필요한 도구들은 모두 소프트웨어 컴포넌트로 구성 및 기술되어 컴포넌트 등록기에 등록되어 저장된다. 등록된 컴포넌트는 GUI 기반의 MDO 커널에 의하여 검색되고 MDO 문제 해결 절차의 구성 요소가 된다. 구성된 문제 해결을 위한 실행은 컴포넌트 서비스 에이전트에 의하여 이루어진다.

EMDIOS는 새로운 설계도구를 EMDIOS에 삽

	b2	b7	이
1	10	10	0
2	10	10	4196 46752981726
3	10 0393999977648	10	4196 46752981726
4	10	10	4196 46752981726
5	10	10	4196 46752981726
6	10	10	4196 46752981726
7	10	10	4196 46752981726
8	10	10 0393999977648	4196 46752981726
9	10	10	4196 46752981726
10	10	10	4196 46752981726
11	10	10	4196 46752981726
12	10 1255423421139	12 4177329036090	4196 46752981726
13	10 3540544446358	17 5340022561767	5179 1543114279E
14	10 3436937652009	17 3901673559874	6769 1761237151

(b) Table

그림 4. Real Time Monitoring

게 통합하여 사용할 수 있도록 공통적으로 구현될 수 있는 부분을 구현한 추상 클래스와 이로부터 필요한 인터페이스를 생성할 수 있는 인터페이스 제조기를 제공함으로써 확장성과 개방성을 제공한다.

제시된 MDO 프레임워크의 사용자 인터페이스는 가장 많은 사용자를 확보하고 있는 윈도우 환경에서 Visual C++를 이용하여 개발되고 있으며, 다양한 OS 환경에서 작동되어야 하는 래퍼는 JAVA로 개발하였다. 현재 개발된 MDO 프레임워크로 간단한 MDO 문제를 설정하여 시스템을 검증하고 있으며, 구체적인 벤치 마크 결과에 대해서는 추후

발표할 예정이다. 본 논문에는 MDO 프레임워크의 전체 시스템 구조에 초점을 맞추었지만, 각 구성 요소에 대한 자세한 내용은 각기 다른 논문을 통해 발표한 예정이다.

이외에 CAD 도구들을 MDO 프레임에 통합하는 연구를 진행 중에 있으며, 분산 환경에서 여러 설계자들이 논의하며 설계를 진행할 수 있도록 지원하는 분산 협동 설계 환경 구축에 대한 연구도 진행중이다. 이들 연구의 결과는 근 시간내에 EMDIOS에 탑재할 예정이다.