

초고온 진공로 설계를 위한 데이터베이스 구축

김종화*, 도상윤*, 이재우**, 정갑주***
건국대학교 산업공학과

Implementing Database for Designing Super High Temperature Vacuum Furnace

* 건국대학교 산업공학과 ** 건국대학교 항공우주공학과 *** 건국대학교 컴퓨터공학과

Abstract

Multidisciplinary Design Optimization (MDO) is an individual and parallel design framework applied in designing large and complex systems. For successful implementation of MDO framework it is essential to manage data in efficient and integrated manner. In this study, we present a case study to implement database to support designing super high temperature vacuum furnace with MDO technology. For that purpose we first extract required data based on the analysis of design process and then data flows between different programs are analyzed. Finally an E-R diagram is presented to design database schema.

1. 서론

1.1 MDO의 기본개념 및 필요성

종래의 순차적인 설계방법은 개념설계에서 최종설계 단계까지 초기설계를 바탕으로 여러 가지 해석을 수행한 뒤, 해석의 결과가 계획된 성능을 만족시키지 못하면 분야별로 시행착오(trial&error method)적인 반복(iteration) 과정을 거쳐서 수정 작업을 수행 후 최종 통합하므로 설계의 효율성이 떨어진다. 그렇기 때문에 설계와 제조기간의 단축 및 비용절감이 어렵게 된다. 그러나 MDO 기술을 활용하면 MDO 프레임워크(frame work)에서 분야간 연계가 자동적으로 이루어지고, 진행되므로 최적화된 설계결과를 적은 비용을 들여서 빠른 기간 내에 도출 할 수 있는 장점이 있다.

MDO(Multidisciplinary Design Optimization)는 대형화되고 복합적으로 구성된 System들을 위한 공식적인 설계방법론 중의 하나로써 정의 할 수 있다. 개별적이고 순차적인 방법론 (sequential methodology)에서 탈피하여 동시적, 병렬적인 방법(concurrent, parallel method)을 적용하여 얻어진 최적설계 결과는 제품의 성능향상과 경제성을 향상시키고 대외적인 경쟁력을 높여줌으로 MDO의 필요성이 점차 강조되고 있다.

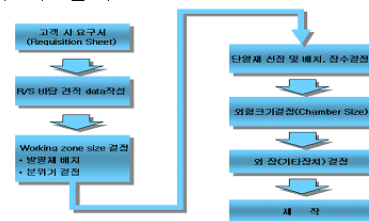
이러한 MDO설계에서 데이터의 공유 및 일관성 있는 관리는 필수라고 할 수 있다. 기존의 순차적인 설계결과를 이용하여 설계하던 방법과는 달리 여러 분야에서의 설계가 동시에 일어나는 데이터들을 관리해야 되기 때문에 데이터의 일관성, 무결성, 최신성등이 요구된다. 따라서 이러한 요구사항을 만족 시킬수 있도록 가용성이 높은 데이터베이스의 설계는 MDO환경에서의 매우 중요한 요소라 할 수 있다.

본 연구에서 MDO기법을 적용하여 연구 및 구현 하고자하는 시스템은 초고온 진공로라는 제품 설계의 지원을 하는 것이다. 진공로란, 원재료 또는 기능요소부품의 정제, 가공, 후처리 공정 등에서 소기의 물리적, 화학적 변화를 달성시키기 위한 초고온의 가열장치를 말하는데, 일반적으로 공업로(爐) 또는 진공로라 한다. 기존의 진공로 설계에서는 순차적인 방법으로 설계자의 경험치에 의존하여 각종 설계변수를 고려한다. 그러나 이 설계의 문제점은 기존의 진공로 설계 시 견적에서 최종제품설계가 될 때까지 필요로 하는 data가 완전히 경험치에 의존되어 왔다는데 있다. 설계자 또한 경험치에 의존한 설계가 과연 적절한 설계인지 검증할 객관적인 방법이 없었다. 이런 필요로 객관적 data에 의한 열 해석 및 물성치 data를 Database화 했다. 또한 기존의 비정형적 업무프로세스를 정형화 시켜 각각의 설계프로세스들을 data flow에 맞게 재구성하여 유효한 data를 이용하여 설계의 최적화를 구현하였다.

2. 설계 요구 data 및 설계 프로세스

2.1 실제 설계에 필요한 data

초기 견적에서 최종설계까지의 업무프로세스는 아래의 그림과 같다.

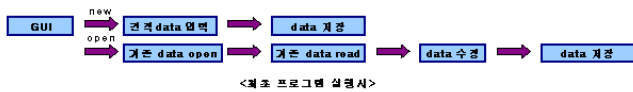


<그림 1> 업무프로세스

고객사에서 발주를 Requisition Sheet(R/S)의 형태로 요구하게 되면 설계부서는 R/S에서 제시된 요구 사이즈에 맞게 견적서를 제시한다. 이 정보를 토대로 장입방식과 사용온도 등을 결정하고, 다음 발열체 배치 및 크기->단열체 배치 및 두께->챔버 사이즈등의 순으로 설계가 결정된다. 초기입력단계 이후, 해석 및 최적화 과정이 있다. 해석만 수행해서 기존 알고리즘을 통해 적절한 데이터를 추출할 수 있고, 이 결과를 이용 최적설계data로 추출해낼 수 있다. 기본적인 수주품별 관리방식은 동일한 Project(동일 제품 하나 마다 프로젝트 ID로 관리)에 설계변수를 조금 바꿔주게 되면 Configuration도 바뀌게 되므로, 고유 Project + 형상 ID로 관리하게 된다. 자세한 부분은 다음 설계프로세스 Flow에서 설명하도록 한다.

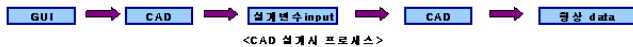
2.2 설계 프로세스 Flow

사용자가 기존의 경험치에 의존했던 설계의 프로세스를 system에 적용할 수 있도록 도시하면 아래와 같다.



<그림2> 견적데이터 입력프로세스

최초 사용자 GUI에서는 견적에서 기본적으로 입력해야하는 정보와 기본형상을 참조해야하는 data 2가지 정보(configuration 테이블)를 입력받는다. 입력된 초기 data를 바탕으로 다음단계의 해석, 최적화등의 단계를 수행한다.



<그림3> CAD입력프로세스

CAD설계는 입력된 configuration data안의 6개의 설계변수를 통해, 각각의 해석, 최적화 단계에서 설계에 적합한 data가 도출되면, 그 데이터를 바탕으로 기본적인 형상을 출력한 뒤, 사용자로 하여금 세부설계를 할수 있도록 한다.



<그림4> DB해석 프로세스

해석단계에서는 초기값에 대한 적절한 범위를 주어 그 범위에 해당하는 데이터를 찾기 위해 DBMS안의 미리 입력된 실험치 데이터를 SQL을 이용하여 최적화 단계로 넘어갈수 있는 최종 결과값을 찾도록 한다.



<그림5> 해석코드 프로세스

해석코드도 해석DB항목과 마찬가지로 프로세스로 진행된다. 해석코드는 GUI상에서 사용자가 연산하고자 하는 정보를 넣게 되면, 미리정의 된 코드연산을 통해 결과 값을 도출하고 그 결과를 저장하게 된다.

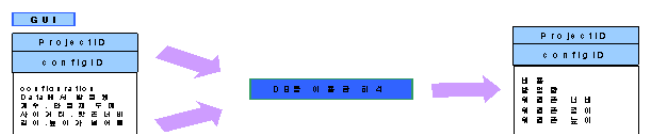


<그림6> 해석DB 최적화 프로세스

최초 고객에게 받은 data를 통해 해당 제품을 설계하기 전 기존 실험에 의한 실제 data를 사용하는 방법이 있다. 그리고 코드라는 정형화된 계산 알고리즘의 방법이 있는데, 해석에는 2가지 방법을 사용하며 각각은 DBMS에 내장되어 있다. 해석DB라고 되어있는 곳은 변하지 않는 데이터를 보간법이라하는 방법을 SQL을 이용하여 찾아내는 방법이다. 그리고 code는 사용자가 임의로 변수를 넣어 알고리즘에 의해 계산된 결과를 도출한 후 해석결과를 나타내는 방법이다. 이후 최적화단계로 넘어가고 마지막으로 CAD도면 및 온도분포(contour)등을 DBMS를 통하여 적절한 값을 도출해서 나타낼 수 있게 한다.

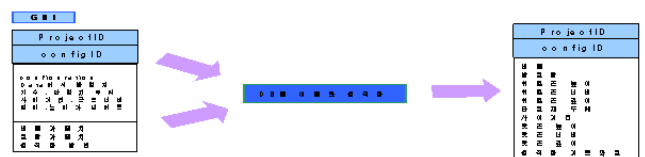
3. 실제 설계 data flow 및 E-R diagram

3.1 data flow



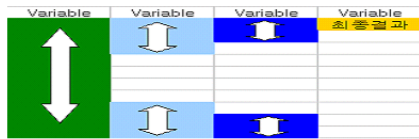
<그림7> 해석DB 입력/출력 데이터

기본 사용자 및 제품정보는 제외하고 크게 최적형상설계를 하기위해 요구되는 data를 도시하였다. 모델에 대한 사이즈를 정한 뒤 설계를 시작하면 Projectid로 DBMS의 기존 정보들을 추출하거나 새로운 Projectid를 부여해서 설계를 시작할 수 있다. 그 아래 형상id는 해석과 같은 다양한 조건을 입력 값으로 넣는데, 동일한 제품에 대해 조건을 다르게 줄 경우 설계형상이 바뀔 수도 있으므로 configid라는 키를 이용한다. 두개의 복합키를 사용함으로써 설계요소 data가 변화 되도 전체 Project에는 변화가 없도록 하였다.



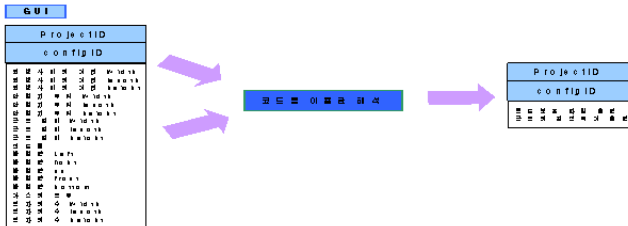
<그림 8> DB최적화 입/출력 데이터

기본적으로 DBMS에 들어있는 데이터를 사용하기 전에 기존의 해석을 하는 방법은 최적화를 하기 위해 원하는 데이터를 FORTRAN 프로그램 안에 배열형태로 집어넣어 전부 비교한 뒤 한번이상의 iteration을 통해 찾는 방법을 사용하였다. 이를 보간법을 이용한 해 찾기라고 한다. 이것을 SQL에 적용해본 결과 간단하게 동일한 결과를 얻을 수 있었다.



<그림9> SQL을 통한 보간법

위의 그림이 각각의 변수에 조건을 부여한 후 보간법이용 해를 찾는 방법이다. 컬럼과 레코드의 크기가 일정한 테이블에서 같은 조건을 주고 찾게 되면 SQL의 비교구문이 길어지기는 하지만 같은 결과를 얻을 수 있다는 결론을 얻었다.

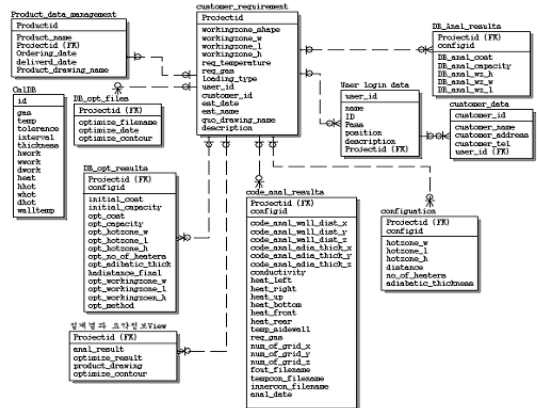


<그림 10> 코드 해석 입/출력 데이터

마지막으로 코드라는 해석을 수행할 때 변수들을 만들어서 어떤 알고리즘에 넣어, 위의 DB안에서 찾는 것과 같은 최적 해를 구하는 다른 방법이 있다. 기본적으로 사용자의 입력으로 data가 만들어진다. 그리고 그 내용이 파일형태로 만들어져 설계해석을 구동하기 위한 다른 응용 프로그램 안으로 들어가게 된다. DBMS에 저장되는 정보는 이 과정으로 출력된 3차원 온도분포의 DATA file과 최종 형상의 높이,너비,깊이의 정보로 이루어진 파일의 경로정보를 담게 된다. 이 정보는 최종단계에서 온도분포 contour 및 설계자로 하여금 최적설계에 알맞은 제품의 크기를 결정하게 하는데 사용된다.

3.2 E-R Diagram 및 데이터 Schema

위에서 나타났던 data flow를 바탕으로 각각의 테이블의 개체관계를 나타내면 다음과 같다.



<그림 11> 설계된 DBMS E-R Diagram

데이터의 개체관계는 위의 그림과 같다. 세부적으로 설명하면 크게 기본정보관련 테이블, 고객정보관련 테이블,설계자 정보 관련 테이블 외 8개의 테이블로 구성된다. 기본정보관련 테이블은 견적서를 받아, 고객사에서 실제 발주가 일어나는 데이터와 발주가 일어나지는 않지만, 발주가 일어났을 때 처럼 해석과 최적화를 하는 프로세스가 있다. 발주가 일어나면 위와 같이 고객요사항 테이블에 제품의 기본정보들을 입력해주고, 이에 따른 기본 configuration data들을 입력해주면 실제 다른 해석 및 최적화 과정으로 들어가기 위한 기본 입력 data로 변환되게 된다. CaIADB라는 테이블은 실제 다른 테이블과는 직접적인 관련은 없다. 해석DB에서 해석결과를 찾기 위한 조건SQL을 이용해 원하는 결과를 얻기위한 실험치 data로 다른 테이블과는 독립적으로 이용되는 테이블이다.

이 개체관계를 바탕으로 통합설계 system의 데이터 Schema를 다음과 같이 구성한다.

customer_requirement					
name	key	type	is null	설명	예제값
Projectid	pk	text	nn	프로젝트 id	20040225
workingzone.shape		text		워킹존 모양	box
workingzone.w		text			
workingzone.l		text			
workingzone.h		text			
req-temperature		text			
req-gas		text			
loadin.type		text			
user.id		text			
customer.id		text			
est.date		text			
est.name		text			
quo.drawing.name		text			
description		text			

<그림 12> 견적테이블 스키마

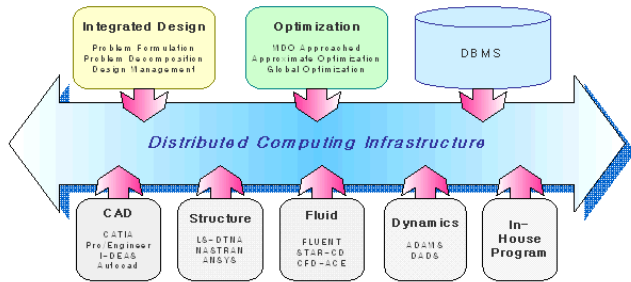
User login data					
name	key	type	is null	설명	예제값
user.id	pk	text	nn	유저 아이디	0001
name		text		워킹존 모양	도상훈
ID		text			mju24
Pass		text			1105
position		text			직원
description		text			원형인공모재각자
Projectid	fk	text			20040225

<그림 13> 사용자 로그인테이블 스키마

테이블에 대한 각각의 Schema정보는 위와같으며, 기본입력정보인 견적데이터와 설계자 기본로그인 정보에 대해서 표현했다.

4. 결론 및 향후 연구방향

MDO환경에 대한 이론적인 그림을 보면 아래와 같다.



<그림 14> MDO Framework 의 구조

아래의 CAD, Structure, CFD-ACE등 어떤 제품에 대한 물성적, 화학적 해석을 한 실제적인 data들을 바탕으로 동시적이고 병렬적인 최종설계가 나오기까지 각각의 모듈안은 database와 연결되어 있다. 이것의 효율적인 사용을 위해 적재적소로 공급해 주기 위한 효율적인 database system의 설계는 MDO환경의 성공의 여부를 결정짓는 핵심요소라 할 수 있다.

본 연구에서는 진공로 설계를 위한 사용자의 요구조건을 바탕으로 데이터 베이스의 설계를 하였다. 설계의 초기단계에서는 사용자의 요구사항에 맞게 데이터베이스에서 관리되어야 할 데이터들을 취합 및 정리했다. 설계 프로세스를 설계하여 실제 system에서 사용자가 설계시에 적용될 여러 단계들을 분석했고, 그에 따른 data flow를 분석해 보았다. 개념적 설계단계에서는 데이터베이스에 저장될 객체와 각각의 속성들을 E-R diagram으로 표현하였고, 논리적 설계단계에서는 실제 테이블에 대한 정규화 작업을 통한 테이블 사상을 거쳐 실제 테이블구축을 하였다. 여기서는 중소기업의 특성을 고려하여 MS-office내 내장된 MS-Access를 사용했다. 규모가 크거나 막대한 data를 관리하기엔 부족한 면이 있으므로, MDO환경을 바탕으로 한 여러 분야에 대한 데이터베이스 system의 구축에는 SQL-Server나 ORACLE등이 이용되어, system이 더욱 많은 가용자원을 가진 데이터 베이스 system을 이용할 수 있으리라 기대된다.

본 연구의 핵심은 통합system 설계를 위한 데이터 베이스 설계과정을 정립하고, 여기에 따른 각각의 프로세스에 대한 분석, data flow정립하였다.

아직 응용개발의 초기이긴 하지만, MDO환경의 활용분야는 무궁무진하고 그에 따른 데이터 베이스의 개발은 중요하다. 본 연구에서 관계형 DBMS를 설계에 사용하였지만, 일반적인 data형이

아닌 해석결과data같은 가변형, 다차원data에 대한 효율적 관리를 위한 객체관계형등의 데이터베이스를 이용하는 방법도 연구해야 할 것이다.

참고문헌

- (1) 김종화, 박우근, “객체지향적 제품데이터 관리 시스템의 모델링”, 건국대학교, 산업공학과 석사학위논문, 2000
- (2) 이재우, 정주영, “다분야 통합시스템 환경 설계 및 적용에 관한 연구”, 건국대학교, 항공우주공학과 석사학위 논문, 2003
- (3) 이재우, 황진용, “시스템 통합최적 설계 환경 구현을 위한 데이터베이스 설계에 관한 연구”, 건국대학교, 항공우주공학과 석사학위논문, 2001
- (4) Ashish Gupta, Sanjai Tiwari, " Distributed Constraint Management for Collaborative Engineering Databases", 1993 ACM 0.s9791-626-319310011 CIKM 'S3 -1 1/93/ D.C , USA
- (5) Neil S. Hall, Robert E. Fulton, " Impact of Data Modeling and Database Implementation Methods on the Optimization of Conceptual Database Design", ASME Engineering Information Mgt. symposium at DETC'97, Sep 14-17, 1997, secremento
- (6) R.Sistla, A.R.Dovi, P.Su, "A distributed, heterogeneous computing environment for multidisciplinary design and analysis of aerospace vehicles", Advances in Engineering Software 31 (2000) 707-716
- (7) Yishai A. Feldman, Jacob Reouven, "A knowledge-based approach for index selection in relational databases", Expert Systems with Applications 25 (2003) 15-37
- (8) 김기성 Visual Basic Database Programming, 삼양출판사, 2000
- (9) 김연홍, 우성미, 문택근, “데이터 베이스 모델링”, 프리렉, 2002
- (10) 문일보, Visual Basic 6 Programming Bible, 정보문화사, 1999
- (11) 홍준호, 김종근, 송건철, 황의범, Oracle Bible Ver 8.x 2nd, (주)영진 출판사, 1999
- (12) Dave Ensor & Ian Stevenson, Oracle Design, O'reilly, 1997
- (13) <http://msdn.microsoft.com/library/>
- (14) <http://www.devpia.com>
- (15) <http://database.sarang.net/>