〈논 문〉

설계공리를 이용한 유리벌브 제품설계 자동화 시스템 구축

도성희* · 박경진**

(1995년 11월 9일 접수)

Software Development for Glass-Bulb Automatic Design Integrated System Using Design Axiom

Sung-Hee Do and Gyung-Jin Park

Key Words: Design Axiom(설계공리), Functional Requirements(기능요구), Design Parameter(설계변수), Mapping Process(사상과정), Design Matrix(설계행렬)

Abstract

As the automation system in manufacturing field works more efficiently, the automation scheme is applied to many areas. In order to reduce the entire manufacturing,cost, the design process must be automated. However, design process is so complicated, it is very difficult to construct the design automation system. The axiomatic approach to design provides a general theoretical framework for all design fields, including mechanical design. The key concepts of axiomatic design are : the existence of domains, the characteristic vecotrs within the domains that can be decomposed into hierarchies through zigzagging between the domains, and the design axioms. Using this approach, the glass bulb design process was analyzed and the design automation software was developed. Through menu display, a user can select or furnish the design input and generate the drawing with ease.

1. 서 론

최근 들어 국내 제조업 분야의 대외 경쟁력을 향상시키기 위해 각 산업체에서는 여러 가지 자동화 시스템을 구축하려는 경향이 지배적이다. 자동화 시스템을 구축하는 방법 및 대상은 여러 가지가 있을 수 있으며, 그 중에서 설계를 담당하는 자동화 시스템의 개발은 제품 생산을 위한 독자적인 기술을 확보하기 위해서 반드시 필요한 대상 중의 하나이다. 그러나 대개의 경우 설계과정은 상당히 복잡하고 어려운 작업으로서 최종 제작 도면을 만들어 내기까지에는 몇 가지 과정을 거치게 된다. 이들 과정을 설계를 수행하는 기술자가 보다 쉽고 빠르게 처리할 수 있는 자동화 시스템을 구축한다면 제품 생산에 상당한 파급효과를 기대할 수 있을 것이다.

설계과정 자체에 대한 학문적이고 체계적인 연구 노력은 설계과정의 일반화가 어렵고 다양하기 때문에 아직까지는 활발한 토의가 진행되고 있지는 않고 있다. 다만 설계공리(design axiom)라는 이론화된 연구 주제가 제기되어 몇 가지 경우에 대해 적용을 시도하고 있다.[1] 설계공리는 설계에 대한 정의 및 이론에 대한 설명을 기능요구(functional requirements, FRs)와 설계변수(design parameters, DPs)로 분석하고 이들의 사상과정(mapping process)을 통해 설계결정과정(decision making process)을 수행하는 일련의 과정으로 구성되어 있

*회원, 한양대학원 기계설계공학과
**회원, 한양대학교 기계공학과

다. 설계과정의 최종 목표는 주어진 조건을 만족하는 좋은 설계(good design)를 창출하는데 있는 만큼 이론적인 배경을 통해 적합한 해(solution)를 찾는 시스템의 구축은 필수적이다. 현재까지 설계공리는 하드웨어(hardware)및 제조과정 등의 설계를 위해 적용되어 왔으며 소프트웨어 설계 분야에는 적용 가능성만이 제시되어 있는 상태이다.[2] 설계공리는 좋은 설계, 적절한 설계변수들의 선정 그리고 가능한 여러 선택 사양들로부터 최적의 해를 찾고자 하는 경우에 있어 합리적인 기준을 제공한다. 따라서 소프트웨어의 설계시 최종 사용자의 의도를 충분히 반영할 수 있으며 소프트웨어 개발 완료 후에도 상당기간 사후관리 등과 같은 업무 발생을 최소화할 수 있다.

본 연구에서는 설계 자동화용 소프트웨어를 개발하는 과정에서 설계공리를 적용하여 보다 효율적인 시스템을 구축하는 방법에 대해 논의하고자 한다. 즉, 설계공리의 관점을 바탕으로 제품설계의 전 과정을 분석하고 분석된 결과를 토대로 제품 설계자가 쉽게 설계를 수행할 수 있도록 설계 자동화 소프트웨어 시스템을 구축하였다. 개발된 소프트웨어의 대상은 TV 브라운관 유리제품(glass bulb)으로서 현재 수행되고 있는 제품설계의 전 과정을 설계공리의 관점으로 재조명하고 효율적으로 개선하는 몇 가지 소프트웨어를 구축하였다. 전체 시스템은 메뉴 시스템을 적용한 사용자 인터페이스(graphic user interface, GUI)기법[3]을 이용하여 사용자가 보다 편리하게 설계과정을 수행할 수 있도록 하였다. 개발된 소프트웨어를 이용하는 경우 제품설계자가 설계공리를 모르더라도 소프트웨어 자체적으로 설계공리를 이용한 분석이 완료된 상태이므로 각종 메뉴를 통해 최적화된 설계작업을 손쉽게 수행할 수 있게 된다.

2. 설계공리에 대한 고찰

2.1 설계공리

대부분의 설계대상은 항상 기능적인 영역(functional domain)에서 언급되는 반면 물리적인 해, 즉 그것을 이루기 위한 방법은 물리적인 영역(physical domain)에서 이루어진다. 설계대상이 되는 특별한 요구들(requirements)에 대한 개념을 기능요구(FRs)라 하고 이들 기능요구들을 만족하기 위한 물리적인 형체는 설계변수(DPs)의 개념으로 특징질 수 있다. 설계과정이란 기능적인 영역의 기능요구와 물리적인 영역에서의 설계변수 사이의 사상(mapping)을 통해 파생되는 과정을 의미하며 이들 두 영역을 관계짓는 것이 바로 설계라고 할 수 있다. Fig. 1은 이러한 관계를 그림으로 표현한 것으로서 고객이 원하는 것이 실제 생산을 위한 변수로 사상되어지는 과정을 설명한다.[4] 즉, 고객이 요구하는 사항을 기능적인 영역내의 기능요구 집합으로 사상시키고 물리적인 영역내에서 기능요구를 만족하는 설계변수를 구한다. 이때 설계변수는 기능요구와 설계변수의 같은 계층구조 레벨을 서로 오고가는 지그재그(zig-zag)방식으로 구할 수 있다. 다음으로 생산변수(process variables, PVs)로 이루어진 생산영역(process domain)으로 사상된다. 이러한 사상과정을 유일한 과정이 아니므로 기능요구에 맞는 적절한 설계변수를 찾아내는 것은 여러 개의 해가 나올 수 있고 많은 부분이 설계자의 손에 달려 있다고 할 수 있다.

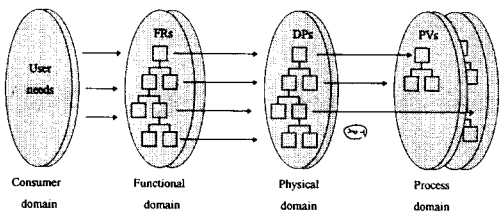

Fig. 1 Concept of domain, mapping and spaces

Fig. 1의 과정을 수행하는데 필요한 지배적인 이론을 설계공리(design axiom)라는 형태로 제공되고 있으며 다음과 같이 정의된다.[1]

제1공리 : 독립공리(The independence axiom)
기능요구 집합(FRs)을 구축할 때에는 각각을 독립적으로 구축한다.

제2공리 : 정보공리(The information axiom)
설계에 필요한 정보량을 최소화한다.

제1공리는 사상과정에서 좋은 설계가 되기 위한 기준을 제공한다. 일반적으로 기능적인 영역과 물리적인 영역 사이의 사상과정에서 설계 방정식은 식 (1)과 같이 나타낼 수 있다.

$$\{FRs\}=[A]\{DPs\} \tag{1}$$

여기서, $\{FRs\}$는 독립적인 FR_i 요소들로 생산물의 기능적인 영역을 표현하는 벡터(vector)이다. 또한 $\{DPs\}$는 $\{FRs\}$의 영향하에서 생산물을 정의

하는 설계변수를 나타내는 벡터이다. $\{FRs\}$과 $\{DPs\}$ 사이에는 설계행렬(design matrix) $[A]$의 곱으로 표현된다. 설계행렬의 인자 A_{ij}는 식 (2)로 구할 수 있다.

$$A_{ij}=\frac{\partial FR_i}{\partial DP_j} \tag{2}$$

이 경우 제1공리를 만족하기 위해서 설계행렬 $[A]$는 대각행렬(diagonal matrix)이나 삼각행렬(triangular matrix)이어야 한다. 설계행렬이 대각행렬인 설계를 비연성설계(uncoupled design), 설계행렬이 삼각행렬인 설계는 비연성화설계(decoupled design)라고 부르며, 이런 설계는 제1공리를 만족한다. 이 외의 다른 설계행렬을 갖는 설계는 연성설계(coupled design)라 한다. 마찬가지로 물리적인 영역과 생산영역 사이의 사상과정은 식 (3)과 같이 나타낼 수 있다.

$$\{DPs\}=[B]\{PVs\} \tag{3}$$

$[B]$행렬은 생산설계행렬(product design matrix)이고, $\{PVs\}$는 생산영역의 생산변수를 나타내는 벡터이다. 똑같이 $[B]$ 행렬의 인자들은 생산변수(process vareiables, PVs)에 대한 설계변수를 편미분한 값이다.

제2공리에서 정의된 정보내용(information content)은 설계의 생산물과 생산과정에 따른 기능요구와 설계변수를 성공적으로 얻을 확률로 정의된다. 정보 I_i는 식 (4)로 정의된다.

$$I_i=\log_2\frac{1}{p} \tag{4}$$

여기서, p는 FRi를 만족하는 DPi를 얻을 확률이다. n개의 기능요구가 있을 때 최상의 설계, 즉 제2공리를 만족하는 설계는 가장 적은 정보 내용을 가진 설계로 식 (5)와 같이 나타낼 수 있다.

$$I_{\min}=\min\left\{\sum_{i=1}^{n}\right\} \tag{5}$$

실제 설계과정에 있어서 제1공리와 제2공리 모두 만족하는 설계를 찾아내기란 쉽지 않다. 대다수의 경우에서는 제1공리를 만족하는 몇 가지 비연성화된 설계들이 제안된 이후에 제2공리를 고려하는 방향이 제시되고 있다.[2,5,6]

2.2 설계공리의 소프트웨어에 대한 적용

소프트웨어의 설계는 1970년대 중반부터 소프트웨어 시스템의 해석과 설계를 체계화시키는 구조화된 기술(structured technologies)의 개발을 통해 괄목할 만한 성장을 이룩하였다. 현재 CASE (computer aided softrware engineering)라는 형태로 제공되는 소프트웨어 설계기법은 요구되는 해석과 설계의 결정과정을 구조적인 접근방식으로 추진하고 있으며 SADT(structured analysis and design technique) 등은 좋은 예이다.[7,8] 구조화된 해석방법은 시스템의 기능요구 사항을 정의하기 위해 상하향식 기능분해(top-down functional decomposition) 방법을 사용하며 단계적으로 접근하는 방식을 사용한다. 그러나 구조화된 해석과 설계기법은 좋은 설계(good design)에 대한 결정 기준(decision-making criteria)을 제공하지는 않는다.

설계공리를 이용한 소프트웨어의 설계는 각각의 설계 목적들을 적절한 설계영역으로 나누고 연속적인 사상과정을 통한 특성 벡터들의 계층구조를 만들어 내는 것으로 나뉘어질 수 있다. 일반적으로 기능적인 영역과 물리적인 영역은 기능요구와 설계변수의 계층적인 구조이며 각 레벨별로 서로의 영역(domain)을 지그재그형태로 넘나들면서 사상을 해 나가게 된다. 소프트웨어 시스템의 설계에 있어서도 기능적인 영역의 기능요구 계층구조와 물리적인 영역의 설계변수 계층구조로 구성되어 진다. 이때 소프트웨어의 결과는 기능요구의 집합이 되고 소프트웨어의 키보드 입력은 설계변수의 집합이 된다. 소프트웨어 코드, 즉 프로그램은 같은 계층구조상에서 설계변수를 기능요구로 변환시키는 설계행렬에 해당된다.[2]

기능요구와 설계변수의 사상과정은 좋은 설계 결과를 얻기위한 방법으로서 물리적인 영역의 최고 레벨의 기능요구에서 시작된다. 이 레벨에서의 설계변수는 제1공리의 독립성을 위반하지 않는 범위내에서 규정된 기능요구와 부합하도록 선택되어진다. 만일 설계해가 선택된 설계변수에 의해서 완전히 만족되지 않는다면 기능요구는 하위 레벨로 분해(decomposition)되어야 한다. 하위 레벨에서 기능요구를 정의할 때 이미 선택된 설계변수는 기능요구를 제한하게 되며, 기능요구의 선정은 명시된 설계변수가 선택되었을 때 제한조건(constraint)을 만족하는 범위 내에서 선택될 것이다. 이러한 방식으로 분해 과정은 수행된다.

설계공리를 이용한 소프트웨어 설계는 다음과 같은 점에서 다른 소프트웨어 개발 방법론과 차이점

을 갖는다. 첫번째로 공리적 접근은 소프트웨어 설계시 네가지 혹은 그 이상의 영역(domain)의 존재를 인정한다. 두번째로 설계과정은 이러한 영역간의 사상을 필요로 한다. 세번째로 각각의 영역은 계층구조를 만들어 가는 분해과정으로 표현되는 특성벡터(characteristic vactor)들을 포함한다. 네번째로 분해과정은 인접한 영역 사이의 지그재그 이동을 필요로 한다. 마지막으로 만족한 만한 소프트웨어 시스템의 생성과정을 검증할 수 있는 사상과정을 포함하는 두 가지 공리가 있다는 것이다.[2]

3. 제품설계 자동화 소프트웨어의 구성

TV나 컴퓨터에 사용되는 모니터의 출력장치로 사용되는 브라운관용 유리는 유리벌브(glass bulb)로 명명되며 그 재질이 유리로 구성되어 있고 제품을 설계하는 과정이 비교적 단순하다. 생산된 제품에는 새도우 마스크(shadow mask)라든지 전자총과 같은 장치가 부착되고 캐비닛(cabinet)에 고정시키기 위한 밴드(band)가 장착되어야 비로소 브라운관이라 부르게 된다. 결국 유리벌브 자체는 하나의 부품으로 취급되는 단위요소 제품으로 전면유리(panel)와 후면 유리(funnel) 두 가지로 구성된다.[9]

현재 제품설계를 수행하는 방식은 데이터베이스의 미비로 생산을 위한 정보 체계가 도면을 통해 이루어지고 있으며 각종 설계과정을 분산된 여러 시스템에서 수행하는 등 비효율적으로 진행되고 있다. 이들을 개선하는 소프트웨어를 제작함에 있어 설계공리를 이용하여 개발되는 소프트웨어의 성능을 향상시키고 각종 설계 레벨에서의 정보 전달 체계를 강화하였다.

3.1 계층구조와 분해

현재 수행되고 있는 제품설계의 과정은 Fig. 2와 같이 진행된다. 이 과정을 기능요구와 설계변수의 사상과정에 도입해 보면 다음과 같다. 우선 기능요구집합을 설정하면

FR1 : 제품의 기초 정보를 구축한다.
FR2 : 제품의 형상을 구축한다.
FR3 : 제품의 성능을 검토한다.
FR4 : 제품 도면을 생성한다.

과 같고 이에 사상되는 설계변수 집합은 다음과 같다.

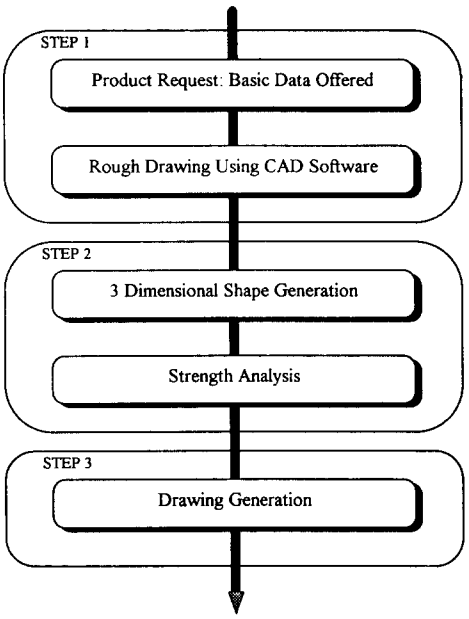

Fig. 2 The design process for glass bulb before decomposition

DP1 : 기초설계 데이터 집합
DP2 : 제품의 3차원 기하형상구조(panel/funnel)
DP3 : 하중 및 외부조건(panel/funnel)
DP4 : 도면 데이터 집합

위의 기능요구 집합과 설계변수 집합의 사상결과인 설계식은 다음과 같이 표현된다.

$$\begin{Bmatrix} FR1 \\ FR2 \\ FR3 \\ FR4 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 & X \\ X & X & 0 & X \\ X & X & X & X \\ X & X & 0 & X \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \\ DP3 \\ DP4 \end{Bmatrix}$$

FR1을 구현하는 방법으로는 거래선에서 제공되는 기초 데이터와 제품 도면을 이용하여 구축할 수 있다. 마찬가지로 FR4를 구현하기 위해서도 기초 데이터, 제품의 3차원 기하형상구조 그리고 제품 도면을 이용하여 구축할 수 있다. 결국 FR1과 FR4는 현재의 설계방식을 따른다면 서로 독립적이지 못하게 되어 위의 설계행렬은 연성화된 결과를 보여주게 된다.

위에서 제시한 연성설계 결과를 개선하기 위해 유리벌브를 설계하기 위한 기능요구 및 설계변수들을 새롭게 고찰하면 다음과 같다. 각각의 경우는 계층화된 구조를 지그재그방식으로 전환해 가며 좋은 설계 방안을 도출해 내게 된다.

step 1 : FRs → DPs(FR을 만족하는 DP 설정)

먼저 최고 레벨의 기능요구집합을 정리하면 다음과 같다.

FR1 : 신제품에 대한 데이터베이스를 생성한다.
FR2 : 제품의 형상을 구축한다.
FR3 : 제품의 성능을 검토한다.
FR4 : 제품 도면을 생성한다.

이에 대한 설계 변수들을 선정하면

DP1 : 신제품 데이터(panel/funnel)
DP2 : 제품의 3차원 기하형상구조(panel/funnel)
DP3 : 하중 및 외부조건(panel/funnel)
DP4 : 악세사리 데이터 집합

과 같으며 이에 대하여 설계행렬은 다음과 같이 비연성화된 설계결과가 생성된다.

$$\begin{Bmatrix} FR1 \\ FR2 \\ FR3 \\ FR4 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 & 0 \\ X & X & 0 & 0 \\ X & X & X & 0 \\ X & X & 0 & X \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \\ DP3 \\ DP4 \end{Bmatrix}$$

여기서, X는 0이 아닌 요소이고 0은 0인 요소로서 X는 프로그램 집합으로 대체할 수 있는 항목이다. DP1과 DP2가 결정되면 FR3은 상용 구조해석 소프트웨어를 이용하여 해결 가능하므로 FR3은 하나의 모듈(M3)로 구성할 수 있다. 나머지 기능요구집합들은 비록 비연성화된 설계일지라도 설계변수 집합의 정보가 명확하지 않으므로 주어진 설계변수를 만족하기 위한 새로운 기능요구를 제안하는 세분화 과정이 필요하다. 이러한 지그재그방식을 통해 설계과정은 분해된다.

step 2 : DPs → FRs(DP를 만족하는 새로운 FR선정)

FR1에 대하여 하위 레벨의 기능요구 집합을 선정하면

FR11 : 신제품의 ID number를 부여한다.
FR12 : 신제품의 데이터 집합을 구축한다.

과 같고 마찬가지로 FR2, FR4에 대한 기능 요구집합을 설정하면 다음과 같다.

FR21 : 곡률을 검토한다. (panel : 편평도/funnel : 축별 profile)
FR22 : 3차원 형상을 계산한다.
FR23 : 제조 가능성을 검토한다.
FR41 : 제품의 형상을 표현한다.
FR42 : 주면 악세사리(accessory)를 표현한다.

step 3 : FRs → DPs(FR을 만족하는 DP 설정)

두 번째 레벨의 기능요구들을 만족하는 설계변수 및 설계행렬을 구하면 다음과 같다. 우선 FR11, FR12에 대해서는

DP11 : 신제품의 대표코드(ssc_code, eiaj code, eiak_code)
DP12 : 신제품의 사양 데이터 집합

$$\begin{Bmatrix} FR11 \\ FR12 \end{Bmatrix} = \begin{bmatrix} X & 0 \\ X & X \end{bmatrix} \begin{Bmatrix} DP11 \\ DP12 \end{Bmatrix}$$

와 같이 나타나며 신제품의 ID 번호는 설계사양 데이터 집합이 만들어지기 전에 수행되므로 비연성화된 설계결과를 도출한다. FR21, FR22, FR23에 대해서는

DP21 : 제품의 내외면 곡률
DP22 : 제품의 기하학적인 연관관계식
DP23 : 금형제작에 관련된 데이터 집합

$$\begin{Bmatrix} FR21 \\ FR22 \\ FR23 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 \\ X & X & 0 \\ X & X & X \end{bmatrix} \begin{Bmatrix} DP21 \\ DP22 \\ DP23 \end{Bmatrix}$$

와 같이 나타난다. FR23을 구현하는 방법중 DP23의 정의가 모호하므로 세분화작업이 필요하다. 따라서 FR23에 대한 하위 레벨에 대한 정의가 다시 필요하다. FR41, FR42에 대해서는

DF41 : 제품설계 데이터 집합
DP42 : 악세사리 데이터 집합

$$\begin{Bmatrix} FR41 \\ FR42 \end{Bmatrix} = \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \begin{Bmatrix} DP41 \\ DP42 \end{Bmatrix}$$

와 같이 나타나며 비연성 설계결과를 도출한다.

step 4 : DPs → FRs(DP를 만족하는 새로운 FR 선정)

FR23에 대하여 하위 레벨의 기능요구 집합을 설정하면

FR231 : 전면 유리의 유효 화면거리(useful screen dimension)를 만족하는가 검토한다.
FR232 : 방출성(ejectability)이 용이해야 한다.
FR233 : 후면 유리의 주사선 편향각도가 만족되는지 검토한다.

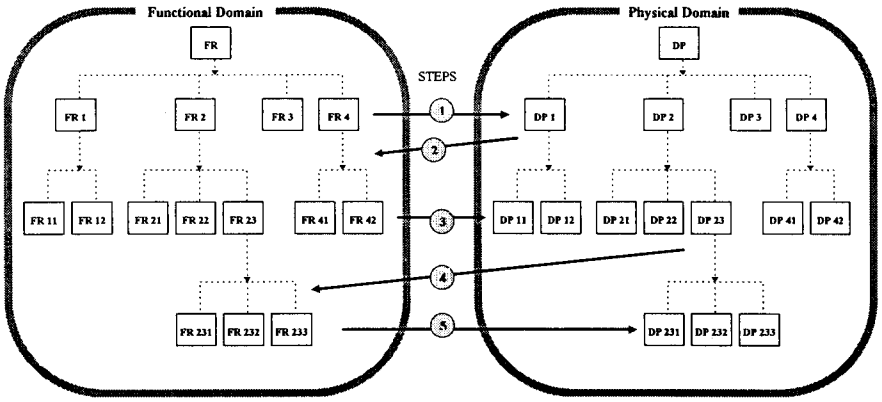

Fig. 3 The zig-zag decomposition of the FR and DP hierarchy structures

와 같다.

step 5 : FRs → DPs(FR을 만족하는 DP 설정)

세 번째 레벨의 기능요구들을 만족하는 설계변수 및 설계행렬을 구하면 다음과 같다.

DP231 : 내면 불렌딩 원(blending circle) 중심점의 중심 축으로부터의 거리

DP232 : 측면 벽(side wall) 각도

DP233 : 요크부 내면 곡률

$$\begin{Bmatrix} FR231 \\ FR232 \\ FR233 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix} \begin{Bmatrix} DP231 \\ DP232 \\ DP233 \end{Bmatrix}$$

FR23을 구성하는 하부 레벨 설계는 비연성 설계결과를 나타내며 결과적으로 FR2을 구성하는 하부레벨 설계는 비연성화된 설계결과를 얻을 수 있다. Fig. 3에 설계용 소프트웨어를 위한 계층구조와 분해과정을 정리하여 설명하였다.

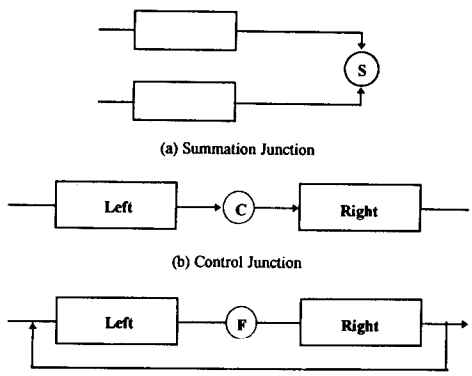

Fig. 4 Module junction type

3.2 모듈결합구조(module-junction structure)

앞에서 제시한 기능요구를 위한 모듈들은 프로그램으로 정형화되고 소비자에 의해 명시된 기능의

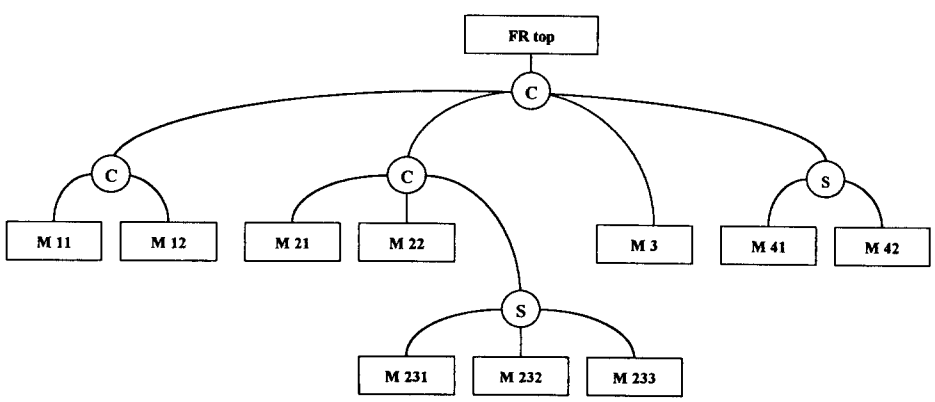

Fig. 5 Module junction structure diagram

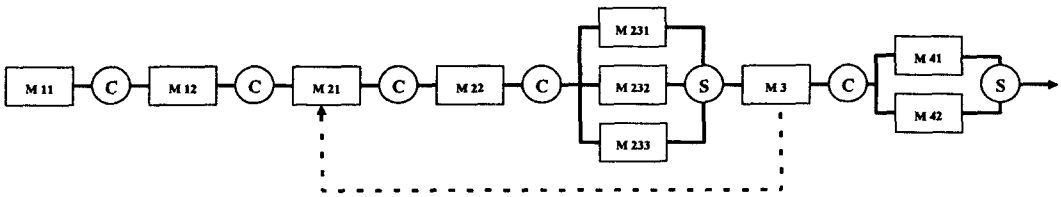

Fig. 6 Information flow diagram

시스템으로 구성되어질 수 있다. 이것은 소프트웨어 설계의 시스템 통합을 의미한다. 각각의 모듈들을 조합하는 방법은 기능요구의 가지들과 그 가지들의 수직 통합을 도표 형태로 표시하는 방법을 이용하며 구체적인 형태로는 결합(junction)과 모듈(module)로 구성된다. 세 가지의 결합형태가 있을 수 있으며 각각의 형태는 다음과 같이 설명할 수 있다. Fig. 4는 결합구조의 형식을 설명한다.

합결합(summation junction : Ⓢ) : 기능요구가 비연성(uncoupled) 되었을 경우로 부모(parent)의 기능요구가 자식(child) 모듈의 모든 결과를 결합함으로써 만족되는 형태이다.

제어결합(control junction : Ⓒ) : 기능요구가 비연성화(decoupled) 되었을 경우로서 부모의 기능요구가 오른쪽 모듈의 행위를 제어하기 위해 왼쪽 모듈의 결과를 사용하는 형태이다.

반복결합(feedback junction : Ⓕ) : 기능요구가 연성화(coupled) 되었을 경우로서 오른쪽 모듈의 결과를 왼쪽 모듈로 되돌려 보내는 형태로 정보의 진행과정이나 프로그램의 이용에 있어서 수많은 반복(iteration)을 요구한다. 반복결합을 포함하고 있는 부모듈(sub-module)이 많을 때 그 프로그램은 곧 감당하기 어렵게 된다.

이들 결합특성을 이용하여 기능요구와 설계변수의 계층적 트리구조(tree structure)와 설계행렬의 형태로 나타난 해석의 결과는 하나의 도표로 나타내어질 수 있다. 이를 모듈결합구조 도표(module junction structure diagram)라고 하며 Fig. 5와 같이 표현한다.[2]

결합특징을 사용함으로써 모듈결합구조도는 모듈들 간의 정보흐름을 보여주는 네트워크형태 그림으로 바뀌어지기가 쉽다. 이는 일반적인 구조화된 해석과 설계방법에서의 중요한 그림적인 표현이었던 정보 흐름도와 같다. Fig. 6은 모듈결합구조도에서 유추된 정보흐름도이다. Fig. 6의 그림중 점선으로 표시된 선은 기능요구와 설계변수들 간의 분해 과정에 대한 반복결합을 의미하는 것은 아니며 단지 설계의 흐름상 해석결과가 좋지 않아 설계변수 값의 변경을 주기위한 반복(iteration)을 수행하기 위한 것이다.

3.3 제품설계 자동화 소프트웨어의 구성

제품설계 자동화 시스템을 구축하기 위한 시스템의 구성은 앞에서 분석한 설계결과의 방법을 충분히 지원할 수 있도록 Fig. 7과 같이 구현한다. 사용자 인터페이스를 통해 사용자는 원하는 기능을 수행할 수 있으며 구축된 데이터베이스는 제품생산의 전과정에 참고 정보로 사용된다.

3.3.1 사용자 인터페이스

모듈결합구조도를 표시한 Fig. 5를 바탕으로 제품설계를 위한 사용자 인터페이스(GUI)를 구성하면, 기능별로 설계변수 입력, 강도해석, 도면생성 등으로 구별할 수 있으며 각각에 대해 전면 유리, 후면 유리를 설계하는 메뉴를 두어 설계작업을 수행하도록 한다.

3.3.2 데이터베이스

설계작업을 수행하는 단계에 있어서 정의되는 각

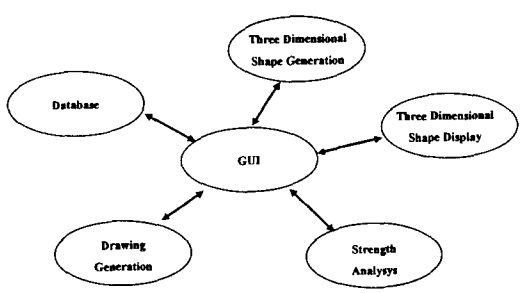

Fig. 7 Structure of the Glass bulb Automatic Design Integrated System (GADIS)

종 설계변수를 관리하기 위해서는 데이터베이스(database)의 구축이 가장 필요한 작업이다. 제품 설계를 위해 필요한 정보는 크게 형상정보와 도면 관련 정보 그리고 관리정보로 분류할 수 있다. 형상 정보는 제품의 3차원 형상에 관한 정보로 각 부위별 치수와 각종 포인트정보(point data), 다항식 정보(polynomial data), 다중 반경정보(multiradius data)등으로 구성된다. 이들 중 일부는 도면을 제작하기 위한 2차원 정보로 변환되어 도면의 형상을 구현하는데 필요하다. 도면 관련정보는 형상정보 외에 각종 측정위치 좌표, 핀(pin)과 같이 제품에 장착되어 부품에 관한 데이터 등의 정보로 구성된다. 관리정보는 제품번호, 설계 진행코드 등이 있다.

데이터베이스로 구축되는 정보들은 끊임없이 입력, 수정, 삭제 등이 작업이 이루어지므로 이들의 관리에는 상당히 주의를 기울여야 한다. 아울러 정보들의 속성을 파악하여 적절한 데이터 군으로 분류하고 이들 군들의 연관관계를 정립하는 과정은 전체 시스템의 성능에 상당한 영향을 미치기 때문에 기능요구와 설계변수의 배치관계처럼 엔티티-관계도표(entity-relationship diagram)를 정의하여 전체 데이터베이스를 구현하는 방법을 많이 사용한다. Fig. 8에 제품설계를 위한 E-R 도표를 설명한다.

3.3.3 3차원 형상생성

유리벌브 제품은 3차원 형상이 중요하기 때문에 주어지는 설계정보를 바탕으로 3차원 형상을 구현하는 소프트웨어가 필요하다. 3차원 형상을 구현하기 위해서는 몇가지 기하학적인 기법을 이용하여 계산을 통해 형상정보를 추출해 낼 수 있다. 기초 설계정보로부터 3차원 형상이 구축될 수 있는지 여부를 판단한 뒤 형상구축이 불가능하면 설계정보를 변경해 가며 형상정보를 구성해 가면 된다.

3.3.4 3차원 형상생성

유리벌브 제품의 해석대상이 되는 작업은 진공해석과 밴드해석의 두 가지가 있으며, 구현된 3차원 형상정보를 바탕으로 유한요소용 메쉬(mesh)를 생성하는 전 처리(post process)과정을 거치면 제품의 강도 등을 검증하는 준비단계가 완료된다. 전처리(post process)과정이 끝나면 강도해석용 소프트웨어를 이용하여 설계된 유리벌브의 강도 해석을 실시한다. 해석결과는 후처리(pre processor)를 통하여 그래픽 적으로 사용자에게 보여질 수 있다.

사용자는 해석결과를 바탕으로 설계변수를 수치값을 조정하는 피드백(feedback) 과정을 거치는 과정이 반복된다. Fig. 6의 정보흐름도 중 점선에 해당하는 과정이 반복(iteration)의 과정을 표시하는 것으로 기능요구와 설계변수를 재조정하는 것이 아니라 설계변수의 값만을 조정하여 전체 시스템을 운영하게 된다.

3.3.5 3차원 형상출력

유리벌브의 설계에 있어서 계산된 제품의 3차원

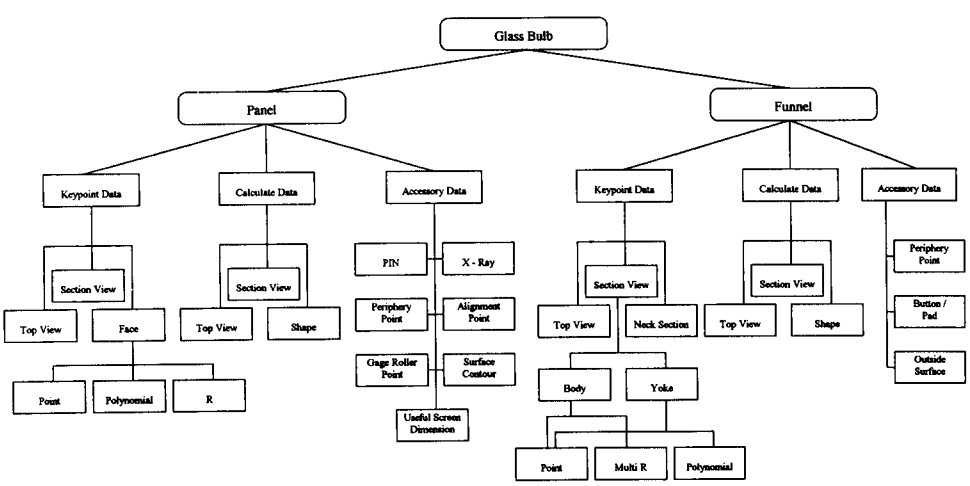

Fig. 8 Entity-Relation diagram for glass bulb database

형상이 정확히 구성되었는지 사용자가 확인해 볼 필요가 있으며, 이들을 위해 3차원 형상의 출력이 필요하다. 아울러 강도해석에 필요한 제품의 메쉬 생성결과 등을 강도해석 이전에 출력해 주는 전처리(post process)로서의 기능도 필요하다.

3.3.6 도면출도

모든 설계과정이 끝나면 설계된 정보를 바탕으로 도면을 출력하는 작업이 필요하며 이 과정은 CAD 소프트웨어를 이용하여 구현할 수 있다. 보통의 경우 도면화 하는 작업은 CAD 소프트웨어에 능숙한 사람이 도면을 제작하게 되지만 설계데이터가 제공된다면 도면화하는 작업 또한 자동으로 처리할 수 있다. 대부분의 CAD 소프트웨어에는 프로그래밍이 가능하도록 연결도구(interface tool)를 제공하며 이들 연결도구를 사용하여 도면을 제작하는 전용 프로그램을 작성할 수 있다.

4. 제품설계 자동화 소프트웨어의 개발

제품설계 자동화를 위한 소프트웨어를 실제로 구현하기 위해서는 3장에서 분석하였듯이 몇가지 성격이 다른 부분들을 통합해야 하기 때문에 제작되는 실행 화일의 크기가 커지는 단점이 있다. 이러한 단점을 보완하고 3장에서 제시한 설계의 분석결과를 적용하기 위해서는 모듈별로 프로그램을 공용 라이브러리(shared library)형태로 만들어 서로 연결하는 방식을 추천한다. 한편, 유리벌브의 설계를 위해서는 수 많은 곡률 데이터 및 포인트(point) 데이터 등을 운영하기 때문에 처리하고자 하는 데이터의 양을 정확히 추정하기가 어렵다. 따라서 이들 정보들을 저장할 메모리 공간을 확보하기가 쉽지 않기 때문에 데이터처리를 위한 기본 방법으로 동적 메모리 할당기법을 사용하여 그때그때 필요한 메모리를 확보하여 사용한다. Fig. 7에 도시된 각각의 기능들을 소프트웨어로 구현하기 위한 방법은 다음과 같다.

4.1 사용자 인터페이스-X-Window/MOTIF

사용자 인터페이스의 도구로는 UNIX 시스템의 표준 그래픽도구인 X-window(MOTIF)를 이용하여 소프트웨어를 개발한다.[10] 사용자 인터페이스를 통해 각종 메뉴 시스템이라든지 그래프 등의 표현 양식을 제공하게 된다. Fig. 9와 Fig. 10에 사용

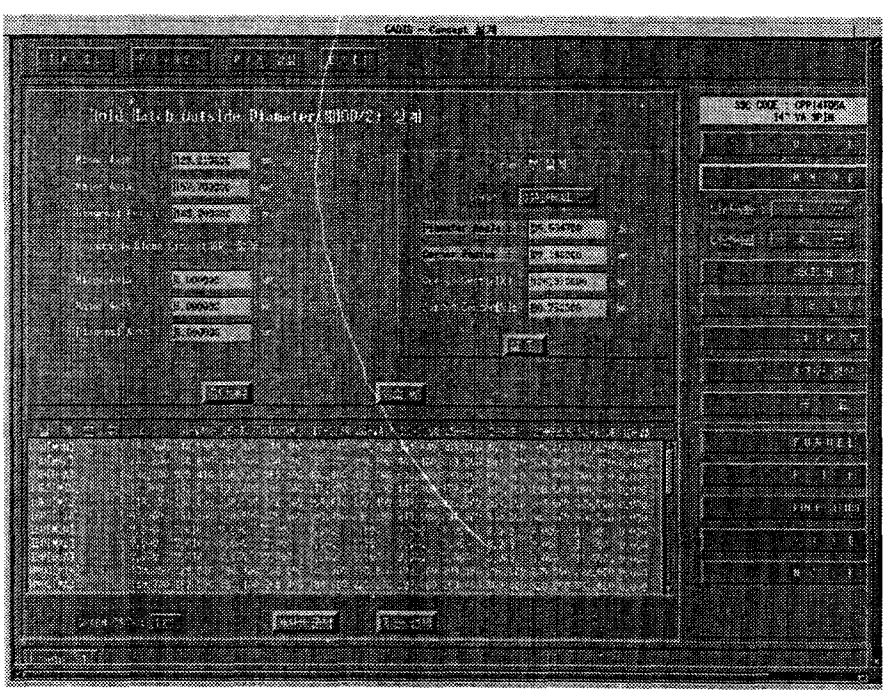

Fig. 9 Sample scene of graphic user interface

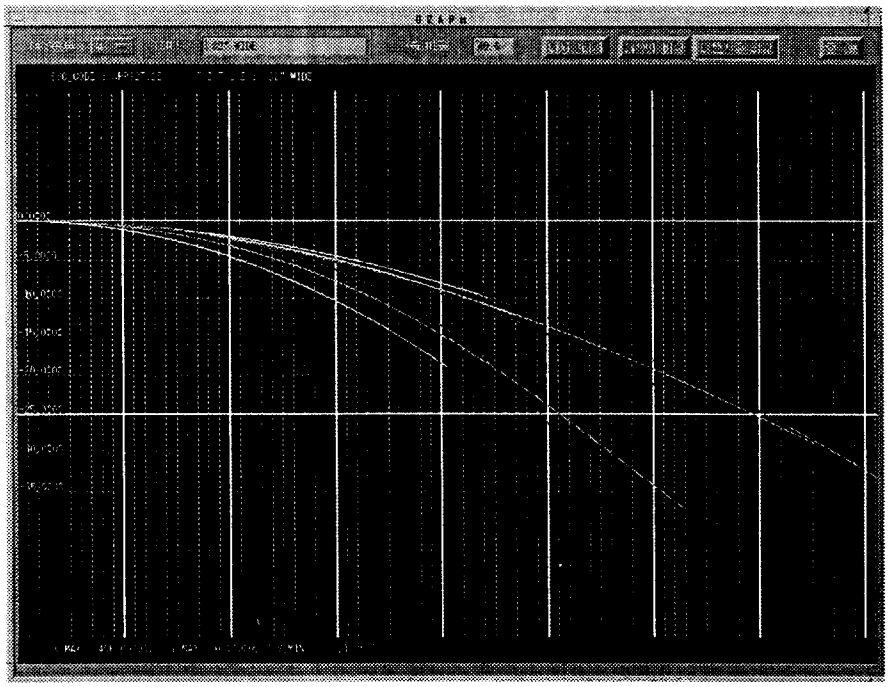

Fig. 10 Sample scene of graph display

자 인터페이스의 한 예를 설명한다.

4.2 데이터베이스-ORACLE

데이터베이스와의 접속을 위한 방법으로는 데이터베이스를 운영하는 소프트웨어를 통해 사용자가 직접 데이터베이스 운영명령을 입력하여 사용하는 방법과 데이터베이스 운영명령을 프로그램화하여 사용하는 방법의 두 가지가 있다. 데이터베이스 운영명령이란 입력, 수정, 삭제 그리고 조회 등에 관련된 명령들로서 이들 명령을 통해 데이터베이스를 구축할 수 있다. 이들 운영명령을 프로그램화하여 사용하는 경우에는 데이터베이스 소프트웨어가 제공하는 컴파일러(precompiler)를 이용하여 실행

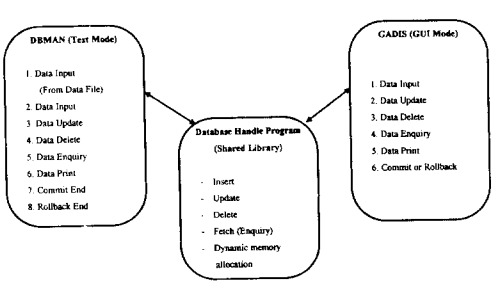

Fig. 11 Structure of the database interface

화일을 제작한 뒤 실행하면 동일한 효과를 얻을 수 있다.[11]

전체 제품설계 시스템의 완벽한 지원을 위해 데이터베이스 접속 프로그램은 두 가지 방식으로 사용하도록 구성한다. 첫 번째는 사용자 인터페이스를 통해 정상적으로 데이터베이스를 접속하는 방법이고, 두 번째는 비상시의 경우(예를 들어 데이터베이스의 복사(backup) 및 이전, 재구축 등)에 그래픽 사용자 인터페이스와는 상관없이 데이터베이스를 접속하도록 하는 방법이다. Fig. 11에 이들의 구성형태를 설명하였으며 접속시 수행되는 기본 명령들은 메모리의 효율적인 이용을 위해 접속방식에 관계없이 공용 라이브러리(shared library)화 하여 사용한다. Fig. 11에서 GADIS는 그래픽 사용자 인터페이스를 이용한 접속을 의미하는 것으로 실제 제품설계 소프트웨어가 동작하면 사용된다. DBMAN은 비상시의 접속을 위해 제품설계 소프트웨어와는 별도로 문자모드로 데이터베이스를 접속한다.

4.3 3차원 형상 및 메쉬생성-BULB-3D

전면 유리와 후면 유리의 3차원 형상 및 강도해석을 위한 메쉬를 생성하기 위해서는 상용 솔리드

모델러를 이용할수도 있으나 3차원 자유곡면으로 구성된 제품형상의 특성때문에 자체 개발된 유리벌브 형상 계산 전용 프로그램인 BULB-3D를 이용한다. BULB-3D는 몇 가지 기하학과 수치 해법을 이용하여 유리벌브의 3차원 곡면형상 및 강도해석을 위한 메쉬를 생성해 주는 역할을 수행한다. Fig. 12는 BULB-3D를 이용하여 메쉬를 생성한 결과를 그래픽으로 보여주는 화면이다.

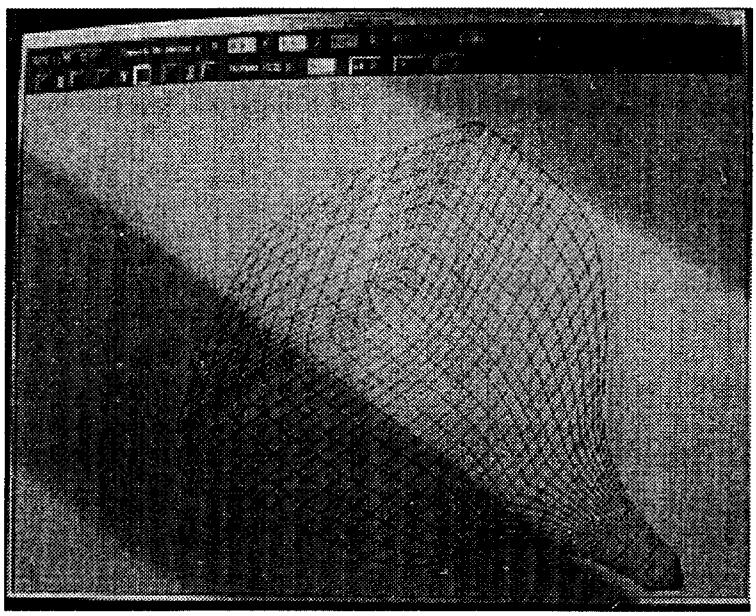

Fig. 12 Sample scene of mesh display

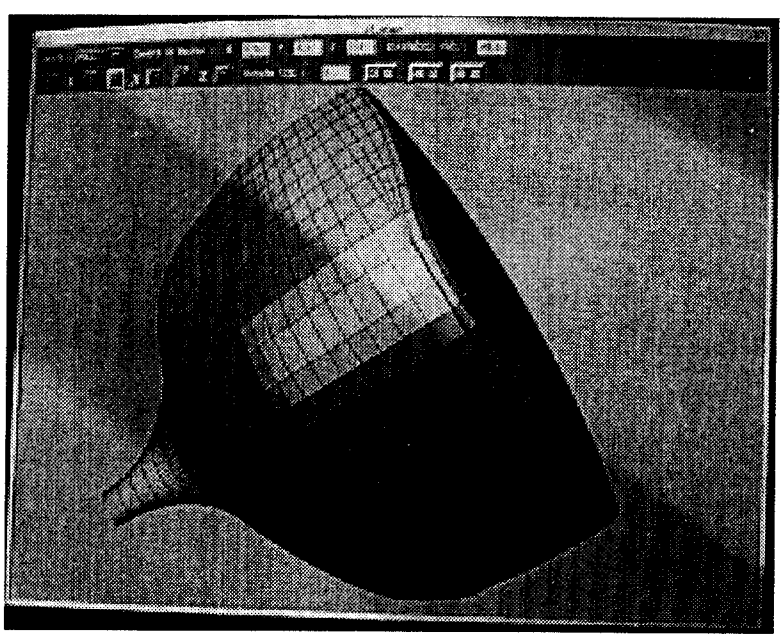

Fig. 13 Sample scene of three dimensional shape

4.4 3차원 형상 출력-Starbase

3차원 형상을 표현하는 기법은 보통 세 가지가 있는데, wireframe형상, 표면형상(surface modeling)그리고 고체형상(solid modeling)이 그것이다. 이들 3차원 형상을 프로그램을 이용하여 표현하기 위한 각종 3차원 지원 그래픽 라이브러리(graphic library, GL)는 몇 가지가 있으며, 대부분 축소/확대(zooming) 기능, 이동(translation) 기능, 회전(rotation) 기능 그리고 랜더링(rendering) 기능 등을 제공한다. 이들 기능 등을 이용하면 제품의 3차원 형상에 대한 검증을 시각적으로 할 수 있다. 참고로 3차원 형상을 출력하기 위한 3차원 그래픽 라이브러리는 스타베이스(starbase)를 사용한다.[12] Fig. 13에 제품의 3차원 형상출력 예를 도시하였다.

4.5 강도해석-ANSYS

강도해석을 위해서는 사용해석 소프트웨어인 ANSYS를 이용하여 제품의 진공강도 및 밴드강도 성능을 검증하게 된다.[13] BULB-3D를 이용하여 메쉬가 정의되었으면 유리의 물성치, 각종 하중조건, 구속조건 등의 데이터를 바탕으로 제품의 강도를 평가한다. 원하는 강도 결과를 얻을 때까지 3차원 형상 생성모듈과 연계해서 반복적으로 수행한다. Fig. 14에 강도해석의 결과 화면을 나타내었다.

4.6 도면출도-Unigraphics

도면출도를 위해서는 유니그래픽스(unigraphics)라는 CAD 소프트웨어를 이용하여 설계된 유리벌브의 도면을 자동으로 출도한다. 유니그래픽스가 제공하는 프로그래밍 연결도구로는 그립(GRIP)과 사용자함수(user function)가 있으며, 사용자함수를 이용하여 C-프로그램만으로 선, 원, 곡선 등을 자동으로 그리도록 할 수 있다.[14] 도면에 나타내는 내용으로는 제품의 형상뿐만 아니라 형상의 치수, 각종 측정위치, 핀장착위치, 정렬위치 그리고 각종 표 등이 삽입되며 사용자의 선택에 따라 도면에 나타내는 항목을 조정할 수 있다. 각종 설계변수에 따라 도면을 자동으로 출도하기 때문에 도면관리가 편리하고 일관성 있게 도면 작성을 할 수 있다. 자동생성된 도면의 예는 Fig. 15에서 볼 수

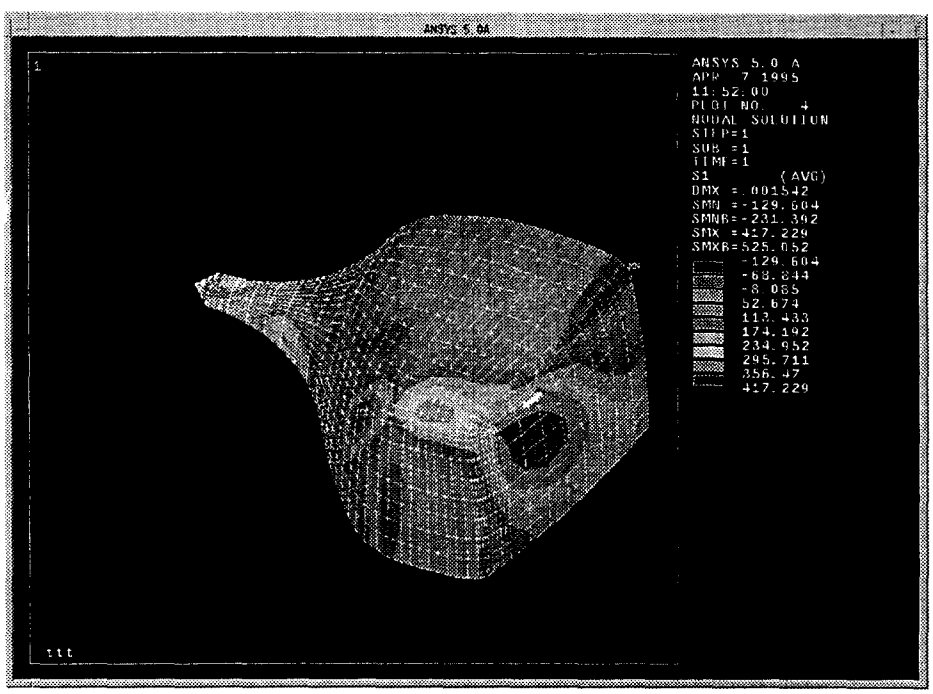

Fig. 14 Sample scene of strength analysis

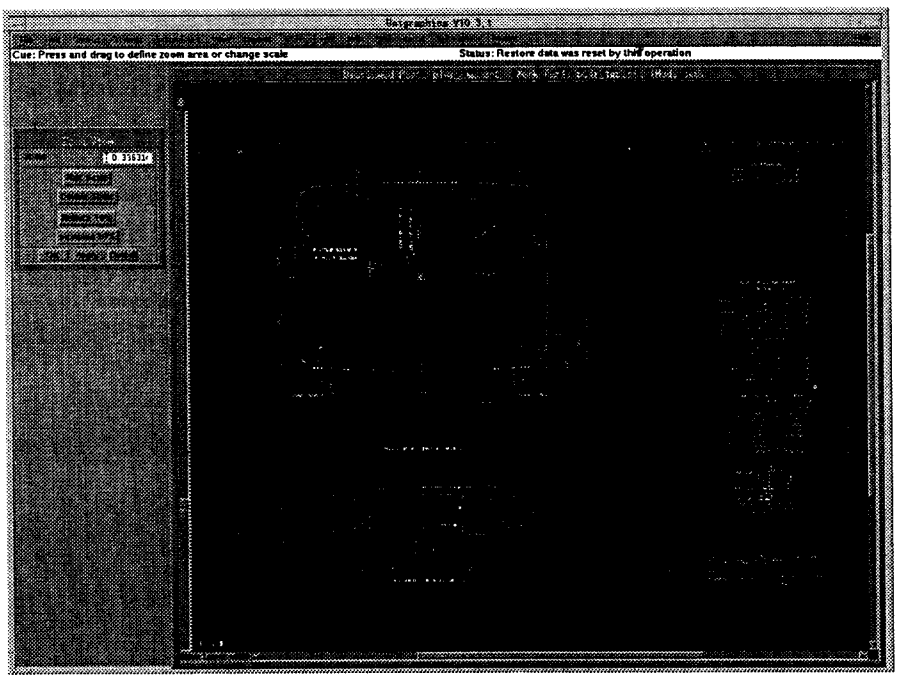

Fig. 15 Sample scene of drawing

있다.

5. 결 론

최근 들어 공학적인 설계(engineering design)에 대한 몇 가지 이론들이 제시되고 있고 설계과정 자체를 개선하려는 일련의 노력들이 진행되고 있는 시점에서 설계과정에 대한 몇 가지 분석을 통해 현재 수행 중인 과정보다 효과적이고 빠른 설계결과를 도출해 내는 방법을 연구하였다. 특히 소프트웨어 설계시에 기존의 구조화된 설계기법이 아닌 설계공리를 적용할 수 있음을 제안하였고, 제안된 방법을 TV용 유리벌브를 생산하는 업체에서 수행하는 제품설계용 소프트웨어 개발에 적용하여 만족할 만한 결과를 얻을 수 있었다. 개발된 소프트웨어를 사용함에 있어서 다른 소프트웨어 개발 방법을 이용하여 개발된 소프트웨어가 없는 관계로 비교 분석을 할 수는 없었지만 설계공리를 이용한 소프트웨어를 가시화하였다는 점에서 의미를 부여할 수 있다고 하겠다.

본 연구를 통하여 자동화 시스템을 구축하려는 추세에 부응하는 일반적인 소프트웨어 설계 방안을 도출하는 하나의 방법을 마련하였다고 생각한다. 개발된 시스템을 통해 신제품의 개발납기 단축, 설계 자립화 기반구축, 제품개발 업무의 간소화 그리고 전 생산공정의 생산 정보단일화 등의 효과를 얻을 수 있었다.

참고문헌

(1) Suh, N.P., 1990, *The Principles of Design,* Oxford University Press, New York.

(2) Kim, S.J., Suh N.P. and Kim, S.G., 1991, "Design of Software Systems Based on Axiomatic Design," *Robotics & Computer-Integrated Manufacturing*, Vol. 8, No. 4, pp. 243~255.

(3) Shiz Kobara, 1991, *Visual Design with OSF/Motif,* Addison-Wesley Publishing, Massachusetts.

(4) Suh, N.P., 1995, "Axiomatic Design of Mechanical Systems," *Tranactions of the ASME-Design Engineering Division*, Vol. 117(B), pp. 2~10.

(5) Albano L.D. and Suh, N.P., 1992, "Axiomatic

Approach to Structural Design," *Research in Engineering Design*, Vol. 4, pp 171~183.

(6) Wallace D.R and Suh, N.P 1993, "Information-Based Design for Environmental Problem Solving," *Annals of the CIRP*, Vol. 42, No. 1, pp. 175~180.

(7) Dart, S., Ellison, R.J., Feiler, P.H. and Habermann, A.N., 1989, "Computer-aided Software Engineering," *IEEE Computer Society Press Technology Series*, Vemuri V.

(8) Ross, D.T., 1985, "Application and Extensions of SADT," *IEEE Computer*, Vol. 18, No. 4, pp 25~35.

(9) 박경진, 도성희, 이정욱, 김창훈, 김재선, 1995, "제품설계 자동화 시스템 구축," Project 최종 보고서, 삼성코닝(주), 수원.

(10) Dan heller, Paula M. Ferguson, 1994, *Motif Programming Manual, O'Reilly & Associates, Inc, Sebastopol.*

(11) *ORACLE 3GL Programmers Guide,* 1990, Oracle Corporation, Redwood Shores.

(12) *Starbase Graphics Techniques,* 1991, Hewlett-Packard Company, Colorado

(13) *ANSYS User's Guide,* 1993, Swanson Analysis Systems, Inc., Houston.

(14) *User Function Programming Manual*, 1993, Electronic Data Systems Corporation, Maryland Heights.