산업디자인 제품화 개발을 위한 정량적모듈의 개발

A Study on the Development of a Quantified Module for the Evaluation of industial Design Proposals

우 홍 룡

서울산업대학교 부설 제품개발 연구소, 연구책임자

산업 디자인 제품화 개발을 위한 정량적 평가모듈의 개발

A Study on the Development of a Quantified Module for the Evaluation of Industrial Design Proposals

우 흥룡* 신 학수 고 을한 한 석우 홍 석기 김 창현
Woo, H. L. Shin, H. S. Koh, E. H. Han, S. W. Hong, S. K. Kim, C. H.

* 연구 책임자, 서울 산업 대학교 부설 제품개발 연구소

ABSTRACT

Design Problems are often both multidimensional and highly interactive. Very rarely does any part of a designed thing serve only one purpose. The activity of designing is thus a goal-directed activity and normally a goal-directed problem-solving activity. This means, problem solving is finding a way to get from some initial situation to a desired goal.

Designers are transforming agents within a society whose goals are to improve the human condition through physical metamorphosis. Many theorist have agreed that designing involves problem solving or decision making. Accordingly evaluation plays an essential role in design activity.

The evaluation factors include all attributes that have levels specified by quantitative and qualitative objectives. Alternatives in multi-objective decision problems generally possess numerous attributes by which they can be described and compared. The evaluation factors include all attributes that have levels specified by quantitative and qualitative objectives. However since qualitative factors are difficult to quantify as numeral estimates, these factors have tended to be ignored without regard for their importance to human content. We adapted the Accumulative Evaluation Model as an evaluation algorithm for IDES.

Industrial Design Evaluation System (IDES) consists of 3 major modules (1.Design Element, 2.Matrix, 3.Evaluation). It is intended to be an aid for design evaluation. The luther thinks IDES is a new design evaluation approach which could provide effective rating of design values to make value judgements. It is an attempt to provide industrial designers with access to design evaluation.

The author's aim is to produce an Object-Oriented Evaluation System which can guide the designers and decision makers under complex design projects.

It uses an Object-Oriented Programming for this prototype, Because of managing complexity (Flexibility and Reusability) and improving productivity(Extensibility & Maintainability and Programming by User) in software development process. The author has chose the C++ because it is a hybrid, rather than pure Object-Oriented Language.

Key Words : Industrial Design Evaluation System(IDES), Computer Aided Industrial Design (CAID), Graphical User Interface(GUI)

1. 서 론

1.1 연구 배경 및 필요성

디자인의 시발점은 '필요(Needs)'이며, '제품'은 그 필요를 충족시키는 하나의 수단으로서 디자인의 도착점이 되는 것이다.[1] 여기에서 필요란 사람들이 문제상황에 직면하여 생기게 된 요구인 것으로, 바로 여기에 그 문제가 풀려져 만족스러운 상태에 도달될 경우의 상황을 염두에 둔 '목적'이라는 것이 존재하게 되는 것이다. 즉 디자인은 한마디로 목적 지향적 문제해결 활동 (a goal directed problem solving activity)으로 볼 수 있는 것이다. 또한 이러한 디자인 문제의 양상은 변화(change)

* 본 논문은 1993년도 한국 학술진흥 재단의 학술연구 조성비에 의한 연구과제임을 밝혀 드립니다.

와 상호작용(interaction)의 원리를 가지고 있으며, 이에 대해 로슨(Lawson)은 "디자인 문제들은 종종 다차원적(multidimensional)이며, 고도로 상관적(interactive)이다."[2]라고 말하고 있다.

한편 디자인 행위를 추상적 일반원리에서 구체적 대상 및 가치를 향하는 일련의 프로세스로 간주할 수 있는데 , 이러한 과정에서 디자인 알고리즘(design algorithm)은 분석(Analysis), 종합(Synthesis), 그리고 평가(Evaluation)의 삼원적 프로세스를 가지게 되는 것이다.[3] 여기에서 한 행위의 목적과 그 행위의 실행에 대한 비교로서, 그리고 실행에 따른 결과의 측정으로서 평가는 디자인 활동의 매우 중요한 부분을 차지한다고 볼 수 있으며, 이를 통하여 디자이너는 종합(Synthesis)에 의해 도달된 특정 제의들(Proposals)의 가치를 결정하게 되는 것이다.

그러나 디자인 프로젝트 과정에서 디자이너들은 '종합(Synthesis)'에 대해서는 상당히 집중을 하는 경향이 있는 반면에 '평가(Evaluation)'에 대해서는 간과하려는 경향이 있다.[4] 즉 평가함에 있어서 디자인 목적에 대한 전반적인 관점을 결핍하는 경우가 흔히 있으며, 다속성에 따른 평정작업의 난해함으로 인하여 합리적 평가의 수행이 거의 불가능하다고 여기는 경우가 흔히 있다. 때로는 이와 같은 평가의 소홀함 또는 편협함으로 인하여 디자인 신뢰도가 하락되는 것은 물론 디자인 실행에 치명적인 결과가 초래되기도 한다.

따라서 본 연구에서는 합리적 평가시스템을 개발하기 위해서 정량적 속성과 정성적 속성을 지니는, 다차원적 디자인 제의에 대한 누적적 평가방법을 고찰하고 그 컴퓨터 어프리케이션을 연구하고자 한다. 이러한 관점에서 표현 위주로 사용되어온 컴퓨터 응용 디자인 작업을 프로세스 전반에 걸쳐 통합시키는 방법론적 접근, 이른바 통합적 CAID (Computer Aided Industrial Design)의 일환으로, 보다 구체적인 디자인 평가모델의 시스템 설계를 연구하였다.

1.2 연구개발의 목표 및 내용

정보화의 추세 속에서 산업 디자인 분야의 컴퓨터 이용은 그 정보형태가 다양화하는 가운데 확산되는 경향을 보이고 있다. 그러나 아직까지 컴퓨터는 대부분 단편적으로 이용되고 있다. 즉 CAD (Computer Aided Design)나 컴퓨터 그래픽스 (Computer Graphics) 및 데이터 베이스(Database) 등과 같은 범용화되고 팩키지화된 상용 소프트웨어의 사용에 그치고 있는 실정이다.

따라서 이제는 각각 별도로 분리된 프로그램이 아니라 전체적인 디자인 프로세스에 통합적으로 적용될 수 있는 디자인 정보처리 시스템이 요구되는 시점에 이르렀다. 또한 각각의 어프리케이션 프로그램에서 생성된 데이터도 객체 지향적 프로그래밍 시스템 아래 멀티미디어 환경과 서로 통합됨에 따라 각각의 어프리케이션 프로그램들과 정보의 호환이 가능하게 되었다. IDES(Industrial Design Evaluation System)은 이러한 추세에 부응하여 디자인 평가 시스템으로 구현됨으로서 디자이너들에 의한 획기적인 평가작업의 개선은 물론, 이로 인해 절약된, 보다 많은 시간을 창조적인 발상에 투입할 수 있게 될 것이다.

본 연구에서는 종합적 디자인 평가 시스템 개발의 필요성에 따라 다차원적인 디자인 속성의 종합평가를 위한 컴퓨터 응용 시스템의 연구개발을 목표로 설정하였으며, 그 세부 내용은 다음과 같다.

1) 정량적/정성적 속성으로 융합된 디자인이 가치 속성을 바탕으로 누적적인 평가기술의 알고리즘을 적용하여 보다 구체적인 종합적인 디자인 평가 시스템을 연구개발 한다.

2) 시스템 환경은, 사용하기 쉽고 효율적인 사용자 인터페이스와 데이터간 손쉬운 전이 방법을 제공하는 장점을 지니는 윈도우 환경을 기본으로 하여, 그 어프리케이션 시스템을 연구한다.

3) 객체지향 GUI(Graphical User Interface)를 적용한 어프리케이션으로 개발하여, 사용자 편에서 신속하고도 용이한 조작이 가능하게 한다.

4) 데이터 베이스의 관점에서, 문자정보 위주에서 그래픽 이미지 등 멀티미디어 정보 디자인에 대비한다. 특히 산업디자인의 데이터의 유형은 시각적인 데이터와 문자정보의 복합적인 구조를 가지는 것으로, 구체적으로는 3-D 이미지(Images), CAD, 2-D 애니메이션(Animation), 멀티미디어(Multi-Media) 등의 이미지 데이터베이스의 적용을 가능하도록 한다.

2. 종합적 디자인 평가시스템의 연구

2.1 종합적 디자인 평가방법

하나의 제품이 완성되기까지의 제품전달 과정 (Product Delivery Process), 또는 제품개발 과정 (Product Development Process)에는 공학적, 그리고 비공학적인 여러 분야의 전문적인 사람들로부터 입력된 정보들이 필요하게 된다. 즉 그와 같은 제품을 연구/개발하는 데는 여러 다른 분야의 전문가들로부터의 통합된 입력이 전제되어야 하는 것이다. 즉 하나의 제품은 제품 디자인의 결과로 등장된 것이며, 여기서 제품 디자인은 그 제품에 영향을 미치는 수많은 기술적인 그리고 비기술적인 구성요소들(성능(Performance), 환경문제(Environment), 인간공학(Man-Machine Interface), 형태미학(Aesthetics) 등)이 누적되어 처리된 것이다.

디자인 평가 시스템을 개발하기 위해서 디자인 가치의 대안별 누적 평가 모델을 채택하였다. 즉 평가항목이 m개 있고 그 중요도로서 척도치(Scale Value) 대신 크리테리아 평정치(Criteria Rating: CR)를 CR_1, CR_2, CR_3 ... CR_i ... CR_m 으로 한다. 다음에 평가대상의 대안이 n 개일 경우 i 번째 평가항목을 바탕으로 우열도를 X_{i1}, X_{i2}, X_{i3} ... X_{ii} ... X_{in}으로 하면, n 개의 대안의 각각의 종합 평가치로서 상대적 가치는 V_1, V_2, V_3 ... V_i ... V_n 을 공식 (2.1)로 나타 낼 수가 있다.[5]

$$V_1 = \sum_{i=1}^{m} X_{i1}\ CR_i \qquad (2.1)$$

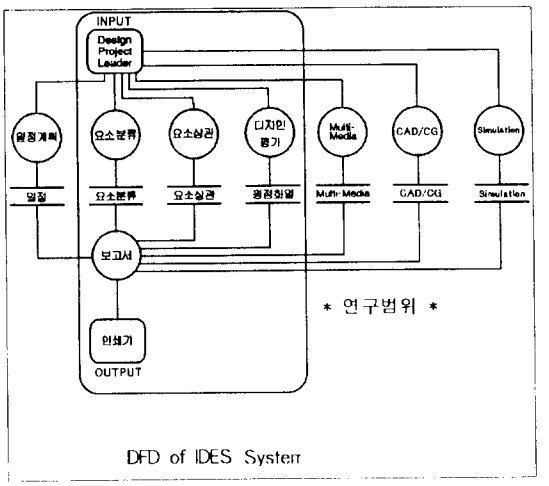
$$\sum_{j=1}^{n} V_j = \sum_{i=1}^{m} X_{ij}\ CR_i = 1.0\ \ (j=1,...,n) \qquad (2.2)$$

여기에서 CR는 대안에 대한 평가항목인 크리테리아 평정치이며, 평가대상인 n 개의 대안에 대한 i 번째 평가항목에 대한 우열도가 Xin 이다. 따라서 n 개의 대안의 누적 평가치($\sum Vj$)는 1.0이 된다. 따라서 대안별 상대평가치의 충화는 공식 (2.2) 과 같으므로, 이에 따라 각각의 대안은 백분율(%) 로서 수량적 가치가 산출된다. 따라서 이와 같은 것들이 서로 균형을 이루지 못한다면 그 제품은 시장에서 실패할 수도 있을 것이다.

토탈 디자인(Total Design)에 의한 구조적 접근이 디자인 성능을 크게 신장시킨다는 것은 사실이다.[6] 기술적/비기술적 대상에 대하여 통합적으로 자료처리가 진행되고, 가시적인 운용이 더해짐으로서 디자인 과정은 보다 효율적이며, 효과적이 된다. 특히 가시성(Visibility)은 통합을 유도하는 중요한 요인으로 볼 수 있으며, 또한 가시성은 여러 사람들이 수행하는 일의 목적과 이유에 대한 이해를 돕는 것이다.

IDES는 시스템 내부적으로 디자인 요소(Design Element), 요소상관(Matrix), 그리고 디자인 평가(Design Evaluation)의 세 가지 주요 모듈로 구성 되어 있으며, 여기서는 디자인 평가 모듈에 논의의 중심을 두고 있다.(Fig.1)

디자인 해결안이 되는 컨셉트(Concept)를 효과적으로 평가하기 위해서 한 조의 인정된 기준들이 필요하다. 크리테리아(Criteria)는 매트릭스(Matrix)를 형성하기 전에 그룹에 의해서 필히 선언되고 동의되어져야 한다. 최근 여러 문헌에서 디자인 행위가 이미 알고 있는, 사전에 이루어진 개념적 선택 또는 수학적 최적화를 가져오기 어려울 수 있는 매개변수들을 내포하고 있는 상황 가운데서 역설적으로 그 최적화를 도모하고자 한다는 사실들을 주제로 다루고 있다. 모든 디자인에는 정량적, 비정량적 매개변수가 존재한다. 그러나 디자인을 수학적(정량적) 용어로만 세부적으로 충분히 표현하기란 거의 불가능하다.



DFD of IDES System

(Fig. 1) IDES System의 평가모듈의 종합적 DFD

그러나 비록 이것이 가능하다 해도 다른 편의 비정량적인 요인들이 쟁점이 된다. 한편 이를 보완하기 위한 물리적 모델링은 수많은 비정량적인 것을 명확화 하는데 도움이 될 것이다. 그러나 여기에도 역시 시간적, 공간적인 문제가 뒤따르게 된다. 즉 디자인 프로세스상 다단계적으로 시행되는 모든 평가의 과정에 일일이 이를 적합 시키기에는 무리가 뒤따르며, 더구나 이의 정량적 평가는 용이하지 않다고 판단되는 것이다.

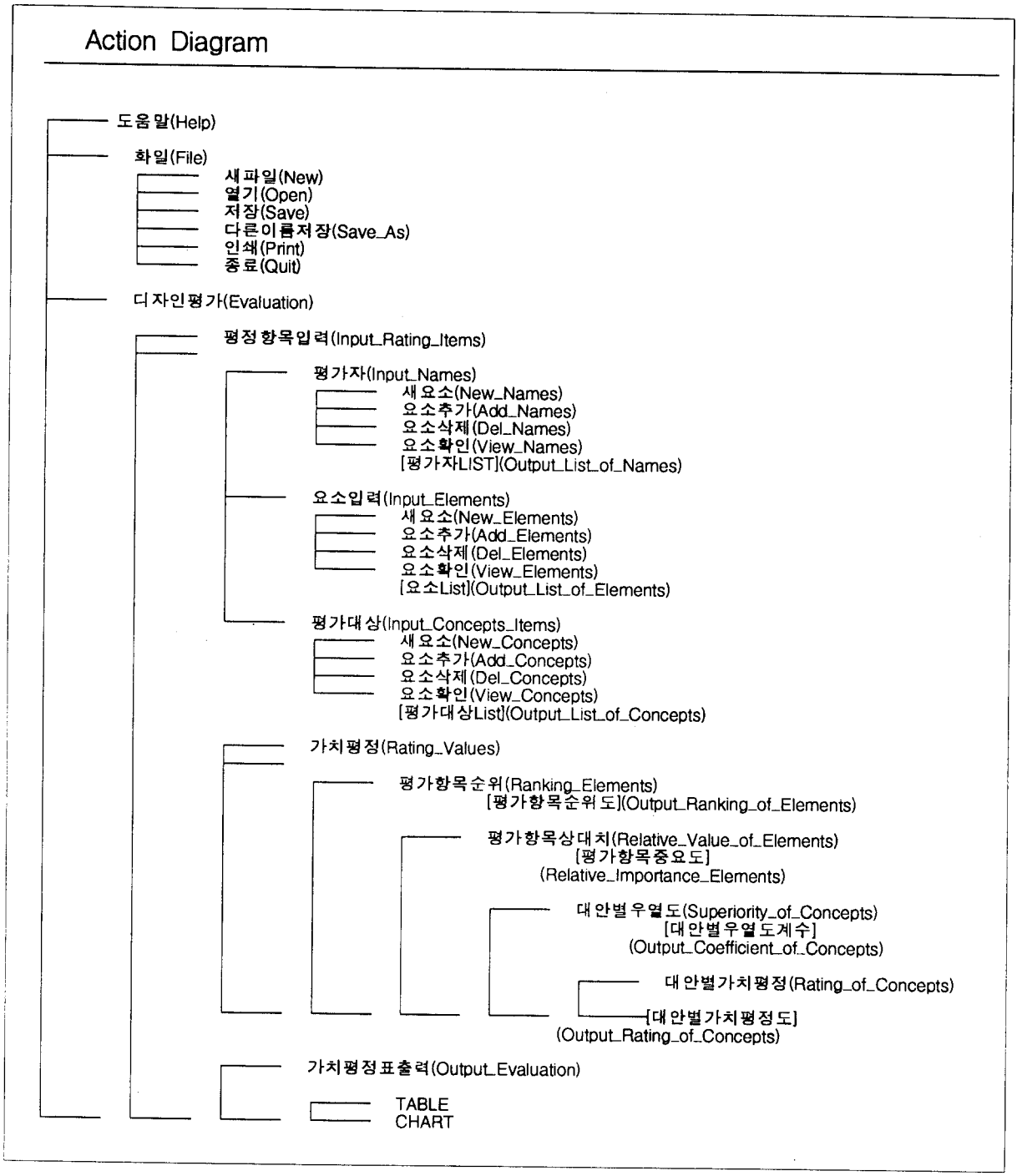
2.2 객체 지향적 시스템의 개발

객체지향이란 문제와 그 해결방법을 가능한 인간의 사고에 가까운 형태로 프로그램하기 위한 개념으로서 실제의 문제영역(problem domain)에 존재하는 사물이 가지는 기능이나 지식을 모델화한 개념적 객체 (conceptual object)이다. 프로그램은 객체의 정의와 그것에 대한 지시(메시지)를 보내는 형식으로 기술된다. '객체'란 어떤 데이터들과 그것을 사용하기 위한 기본적인 조작, 절차들을 하나로 하여, 하나의 모듈로 간주한 것이라고 보는 것이 가장 넓은 해석이다.

여기에서 공통적으로 취급할 수 있는(공통의 메소드를 갖는) 객체는 하나의 크래스(class)로 정리한다. 크래스로 정리된 객체는 그 각각이 그 클래스의 '인스턴스(instance)'라 부른다. 그리고 각각의 객체(인스턴스)가 가지고 있는 메소드는 크래스의 인스턴스 모두에 공통적인 메소드로서 크래스를 가지고 있게 된다.[7]

객체지향 시스템의 대표적인 특성으로 추상화는 복잡한 문제 또는 단순하면서 대규모적인 문제를 간결하게 취급하여 본질을 취하기 쉽게 된다는 것이다. 따라서 이를 통하여 독립성이 높으며 범용성이 있는 모듈을 생성하게 된다.

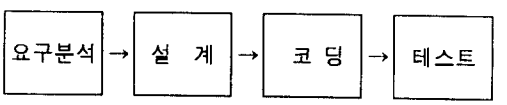
이와 더불어 모듈사이의 정보(데이터와 기능)의 상호작용에서 최소한의 정보만을 보이고, 그 이외의 자세한 것은 외부



(Tab.1) Action Diagram

의 모듈로부터 액세스 할 수 없도록 감추어 버리는 이른바 정보은폐(information hiding)에 의해서 데이터 및 그 데이터를 조작하는 기능(함수)을 하나의 모듈에 집어넣은 것이 된다. 이것에 의하여 외부로부터 는 불필요한 데이터와 불필요한 함수가 감추어 져서 모듈의 독립성이 높아져서 필요한 데이터와 기능에만 주의를 기울일 수 있게 된다.. 이는 데이터와 그 조작을 일원화시킨 '추상 데이터형' 과함께 프로그램의 추상화 레벨을 더욱 높이게 되는 것이다.

객체 지향적 평가시스템의 개발은 전형적인 소프트웨어 라이프사이클의 공정에 따라 요구분석(requirement analysis), 설계(design), 코딩(coding), 그리고 테스트(test)의 단계로 추상적인 개념에서 출발하여 구체화로 진행하였다. 단 여기서 운용과 보



(Fig.2) 시스템의 라이프싸이클 [8]

수(operating & maintenance)의 단계는 생략하여 프로토타이핑(prototyping)으로 작성하였다. (Fig.2)

1) 니즈분석

사용자의 니즈를 분석하여 객관성을 가진 요구(requirement)로 정리하는 과정으로, 산업 디자인 개발업무의 내용, 규모를 감안하여 작성하였다. 니즈분석에 따라 요구사양서를 작성한다.

이를 위해서 DFD(Data Flow Diagram)을 적용하였다. DFD는 사용자 업무의 발주 데이터를 좌측에서 우측으로 향하여 데이터 변환 공정을 거쳐서 출력원으로 흘러가므로 실세계의 데이터를 그대로 모델화 할 수 있다는 장점이 있다. DFD는 정보처리에 임하는 사용자의 요구를 넓은 시각에서 분석하는 매우 편리한 도구일 뿐만 아니라 설계의 기본이 된다는 점에서 가장 많이 사용하는 표기법이다. 즉 자료의 입력.처리.저장 과정을 분석해 DFD로 그리는 이유는 그 DFD 자체가 소프트웨어 정보가공체계를 설명하는데 매우 유익하기 때문이다. 또한 초기의 DFD는 요구 분석이 진행되는 중에 한 절차가 상세 분석되면 또 다른 DFD를 생성시킨다.

요구사항의 명세화는 (Tab.1)과 같이 Action Diagram으로 작성되었으며, 그에 따른 Mini Spec.은 (Tab.2)와 같이 정리되었다.

Mini Spec. of IDES

•. IDES Logo 화면이 Windows상에 디스플레이 된다.

- 1분간 화면에 디스플레이, 또는 아무 키의 입력으로 Main Menu 화면으로 바뀐다.

0. Main Menu 화면이 디스플레이 된다.

- Pull Down Menu와 함께 Icon Type의 Tool Bar가 설치되어 마우스 또는 화살표 키로 1.~6.의 해당 메뉴를 선택한다.

1. About IDES를 화면상에 출력한다.

2. 도움말(Help)을 화면상에 출력한다.

3. 프로젝트(Project) 화면에 Pull Down Menu와 함께 Icon Type의 Tool Bar가 설치되며, 3.1 ~ 3.6의 하위 메뉴가 디스플레이 된다.

3.1 새프로젝트 (New_Project)를 해당 Directory에 입력한다.

3.2 프로젝트 읽어오기 (Load_Project)으로 이전 프로젝트의 읽어오기는 물론 특히 부분적인 Block의 Read/Write를 할 수 있다.

- 해당 프로젝트 전체의 Delete기능을 확인과 함께 가능하게 한다.

3.3 저장(Save_Project)으로 프로젝트의 Data를 해당 Directory에 입력한다.

3.4 다른이름저장(Save_Project_As)으로 프로젝트의 Data를 해당 Directory에 다른이름으로 입력한다.

3.5 인쇄(Print)로 Windows의 기본 인쇄기능을 이용하여 프린터 출력을 한다.

3.6 종료(Quit)로 IDES를 종료하고, 저장여부를 확인 한 뒤, Win Main으로 나온다.

6.디자인평가(Evaluation) 화면에 Pull Down Menu와 함께 Icon Type의 Tool Bar가 설치되며,

6.1 ~ 6.3의 하위 메뉴가 디스플레이 된다.

6.1 평정항목입력(Input_Rating_Items)으로 하위 Pull Down Menu 6.1.1 ~ 6.1.3이 설치된다.

- 평정항목입력이 끝나면 화면은 상위 화면으로 바뀐다.

6.1.1 평가자(Input_Names)로 하위 Pull Down Menu 6.1.1.1 ~ 6.1.1.5가 설치, 결과는 6.1.1 화일에 보관된다.

6.1.1.1 이름등록(Input_Names)으로 성명, 부서, 직위, 전화번호가 6.1.1 화일에 입력

6.1.1.2 이름부르기(Load_Names)

6.1.1.3 이름삭제(Del_Names)

6.1.1.4 이름확인(View_Names)

6.1.1.5 [평가자LIST] (Output_List_of_Names) 화면에 Table로 출력

6.1.2 요소명(Elements_Names)에 하위 Pull Down Menu 6.1.2.1 ~ 6.1.2.5가 설치, 결과는 6.1.2 화일에 보관된다.

6.1.2.1 요소등록(Input_Elements)으로 요소명이 이미지 화일과 함께 6.1.2 화일에 입력

6.1.2.2 요소삭제(Del_Elements)

6.1.2.3 요소확인(View_Elements)

6.1.2.4 [요소List] (Output_List_of_Elements) 화면에 Table로 출력

- View Graphic으로 해당요소를 디스플레이 한다.

6.1.3 평가대상(Input_Concepts_Items)에 하위 Pull Down Menu 6.1.3.1 ~ 6.1.3.5가 설치, 결과는 6.1.3 화일에 보관된다.

6.1.3.1 평가대상등록(Input_Concepts)에 Concept, 제조자, Model No 가 이미지 화일과 함께 6.1.3 화일에 입력된다.

6.1.3.2 평가대상삭제 (Del_Concepts)

6.1.3.3 평가대상확인 (View_Concepts).3 화일에서 확인한다.

6.1.3.4 [평가대상List] (Output_List_of_Concepts)로 Concept, 제조자, Model No 등을 화면에 Table로 출력한다.

6.2 가치평정(Rating_Values)으로 하위 Pull Down Menu 6.2.1 ~ 6.2.4가 설치,결과는 6.2 화일에 보관된다.

- 가치평정의 입력이 끝나면 화면은 상위 화면으로 바뀐다

6.2.1 평가항목순위(Ranking_of_Elements)로 입력된 요소입력을 바탕으로 일대비교하여 중요도에 따라 판정하여 점수 및 순위를 산출한다.

6.2.1.1 [평가항목순위표] (Output_Ranking_of_Elements)로 6.2.1의 결과를 Table로 작성 출력한다.

6.2.2 평가항목상대치(Relative_Value_of_Elements)로 앞에서 작성된 평가항목 순위를 바탕으로 비교하여 상대치와 중요도를 산출한다.

6.2.2.1 [평가항목중요도](Output_Relative_Value_of_Elements)로 6.2.2의 결과를 Table로 출력한다.

6.2.3 대안별우열도(Superiority_of_Concepts)로 대안별로 각각의 평가항목에 상대치와 우열도 계수를 산출한다.

6.2.3.1 [대안별우열도계수] (Output_Coefficient_of_Concepts)로 6.2.3의 결과를 Table로 작성 출력한다.

6.2.4 대안별가치평정 (Rating_of_Concepts)로 중요도 계수와 대안별 우열도 계수의 계산으로 대안별 가치평정을 한다.

6.2.4.1 [대안별가치평정표] (Output_Rating_of_Concepts)로 6.2.4의 결과를 Table로 작성 출력한다.

6.3 가치평정표출력 (Output_Evaluation)로 하위 Pull Down Menu 6.3.1 ~ 6.3.3이 설치

6.3.1 [TABLE_Evaluation] Project별로 대안별 종합평가치를 Text모드로 출력한다.

6.3.2 [Table_Elements_Evaluation] Project별에 따라 평가 항목별로, 중요도순으로 각각의 대안에 나타낸 Table로 출력한다.

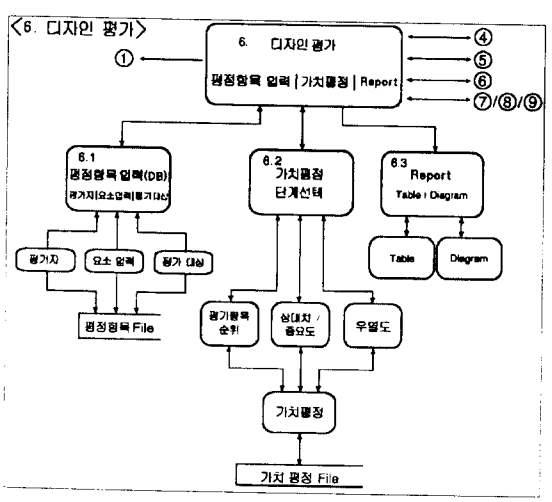
6.3.3.[CHART_Evaluation] Project별로 Pie Chart와 Radar Chart가 작성 출력, Graphic 모드로 출력한다.

(Tab.2) Mini Spec. of IDES

2) 시스템 설계

업무 시스템의 요구 사양서를 받아서 하드웨어(기능) 구성과 소프트웨어(데이터 기능) 구성을 검토하여 컴퓨터 시스템 요구사양서를 작성하는 것이 일반적인 과정이다. 즉 개요설계를 수행하여 입출력 등의 기본적인 방법을 결정하고 각각의 구성요소가 무엇을 할 것인가를 기술하는 일이다. 개요설계에 따라 프로그램 설계 즉, '프로그램 구조설계'와 '모듈 내부설계'를 한다. 이 단계에서 각각의 프로그램을 프로그램 단위로 독립된 기능 단위인 같은 모듈로 분해하여 모듈의 집합으로서 계층 구성화 한다.

설계의 표현은 DFD를 적용하였다. 즉 상위 레벨에서의 데이터 플로우 다이어그램은 비지네스 처리와 그 처리의 결과 생기는 데이터를 표시하기 위하여 사용한다. 하위레벨에서는 프로그램 또는 프로그램 모듈과 그들 루틴 사이의 데이터 흐름을 표시하는데 사용한다. (Fig.3)



(Fig.3) DFD of Evaluation Module

IDES 자료항목 구성표

1. 평가자 List (4.1.5 Output_List_of_Names) =
Project + { No + Name + Depart + Position + Tel }^{n}
2. 요소 List (4.2.5 Output_List_of_Elements) =
Project + { No + Element + {Name a + b + c ...} + [s|g] }^{n}
3. 요소분류표 (4.4.1 Table_Elements [5W1H|M/M/E] =
Project + { No + Element + C1 + C2 + C3 + ... }^{n}
4. 평가자 List (5.1.5 Output_List_of_Names) =
Project + { No + Name + Depart + Position + Tel }^{n}
5. 요소 List (5.2.5 Output_List_of_Elements) =
Project + { No + Element + {Name a + b + c ...} + [s|g] }^{n}
6. Matrix Table(5.4.1 Table_Matrix) =
Project + {No + Element + Marks + Ranks }^{n}
7. 평가자 List (6.1.1.5 Output_List_of_Names) =
Project + { No + Name + Depart + Position + Tel }^{n}
8. 요소 List (6.1.2.5 Output_List_of_Elements) =
Project + { No + Element + {Name a + b + c ...} + [s|g] }^{n}
9. 평가대상 List (6.1.3.5 Output_List_of_Concepts) =
Project + { No + Concept + Manufacturer + Model_No + [s|g] }^{n}
10. 평가항목 순위표 (6.2.1.1 Output_Ranking_of_Elements) =
Project + { Rank + No + Element + { Name a + b + c ... } + Total }^{n}
11. 평가항목 중요도 (6.2.2.1 Output_Relative_Value_of_Elements) =
Project + { Rank + No + Element + { Relative_Value a + b + c ...} + { Importance a + b + c ... + G } }^{n}
12. 대안별 우열도 계수 (6.2.3.1 Output_Coefficient_of_Concepts) =
Project + { Rank + Element + { Concepts +
{ Relative_Value a+ b + c +... } + { Superiority a + b + c +...+ G }}^{m} }^{n}
13. 종합평정표/평정점수(6.3.1 Table_Evaluation) =
Project + { Concept a + b + c ...} + Evaluation + {Name a + b + c ...} + { Element a + b + c ...}
14. 종합 평정표/요소별 (6.3.2 Table_Element_Evaluation) =
Project + { Element + Importance + { Concept a + b + c ...} }^{m} + { Name a + b + c ...} ^{n}

(Tab.4) IDES 자료항목 구성표

구조화 분석단계에서 DFD를 작성할 때는 시스템의 추가 정보를 제공하는 목적으로 처리명세서(process specification)와 함께 데이터 사전(data dictionary)을 작성한다.

자료사전은 DFD의 대상이 되는 자료의 내용을 구체적으로 명시한 것이다. 시스템의 정보영역을 완전하게 파악하려면 DFD와 DD는 공존해야 한다. DD에 삽입되는 각 자료는 자료항목(field), 어트리뷰트(attribute), 자료개체(data entity)등으로 불려지는바, DD의 작성은 화일의 레코드(record)를 정의하는 것으로 볼 수 있다.[9] (Tab.4)

설계작업은 분해와 통합이다. 프로그래밍에서는 근본을 '주모듈(main module)' 분할된 각각을 '하위모듈'이라 부른다. 여기에서 절차에 주목하여 계층구성하는 'top-down 방법'과, 앞에서 정의된 하위 모듈을 통합하여 부분 프로그램으로부터 전체로 정리하여 가는 'bottom-up 방법'을 절충하여 진행하였다.[10]

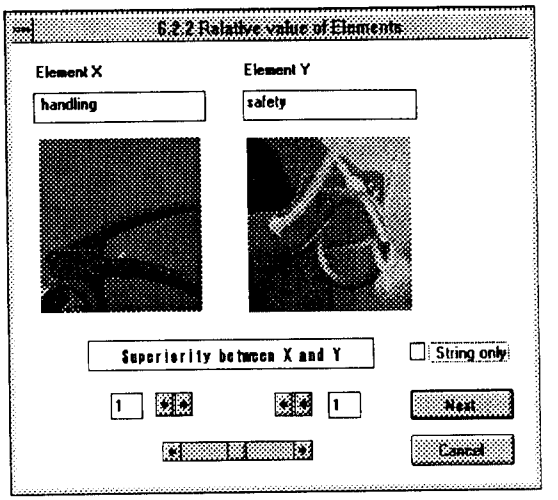
사용자 인터페이스는 풀다운 /팝업 메뉴와 다중메뉴를 복합적으로 적용하고, 이와 더불어 그래픽 사용자 인터페이스(Graphic User Interface:GUI)를 적용하여, 배움의 시간(learn time), 수행속도(speed of performance), 사용자 오류율(rate of errors by users), 기억력(retention over time), 그리고 만족감(subjective satisfaction) 등의 HCI (Human Computer Interface) 관점의 효율성을 고려하였다.[11]

3) 코딩

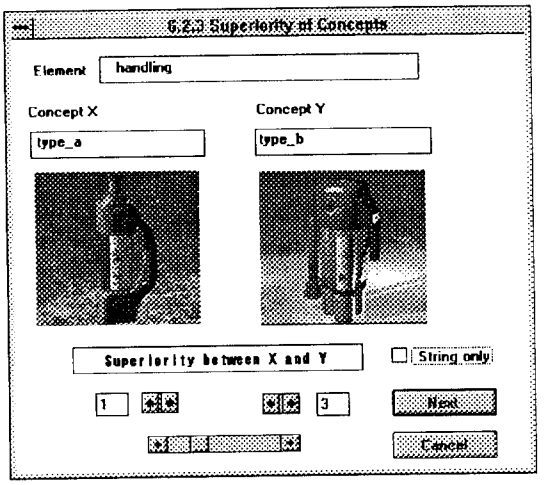
프로그램 사양서에 의해 컴퓨터에서 동작하는 프로그램으로 변환하는 단계이다. 여기에서 모듈 내부 설계를 기초로 운영체제(OS)의 언어처리계에서 기계어로 번역 가능한 언어 즉, 객체지향 프로그래밍 언어로서 C++과 OWL (Object Windows Library)를 적용하여 원시 프로그램(source program)을 작성하였다.

(Fig.4)-(Fig.7)은 가정용 소화기 디자인 제의에 대한 평가를 Project 사례로 적용하여 평가 시스템에 적용한 결과의 한 사례이다. 항목별 평정과정에서 대상의 이미지(Images)가 문자정보(Alpha-numerics)와 함께 출력(display) 되어 신속하고 용이한 평정이 가능한 윈도우 어프리케이션으로 개발하였다. 특히 OWL의 응용 프로그램으로서 대화용 상자(Dialogue box)를 활용하였으며, 여기에 표준 사용자 인터페이스로 Top Down Menu와 Icon Bar를 복합적으로 적용하여 제어가 가능하도록 하였다.

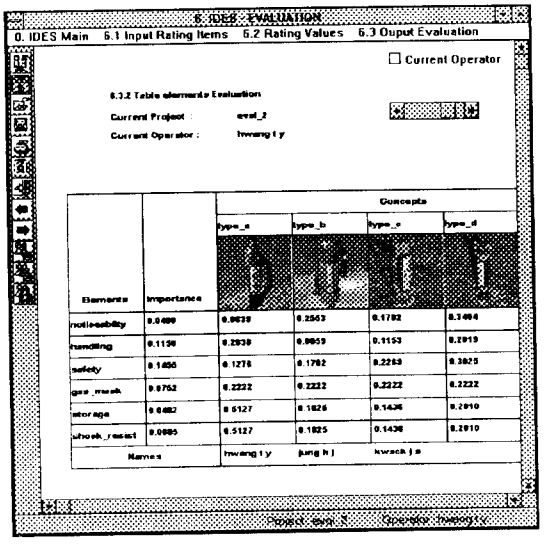
이를 위해 제어 크래스(Control Class)로서 1) List box, 2) Scroll bar, 3) Button, 4) Check box, 5) Radio button, 6) Group box, 7) Edit control, 8) Static control, 9) Control box 등이 응용되었다. 특히 각각의 대화용 상자에는 비트맵(Bitmap) 이미지를 삽입하여, 수직 수평 이동, 확대/축소가 가능하도록 하였다.



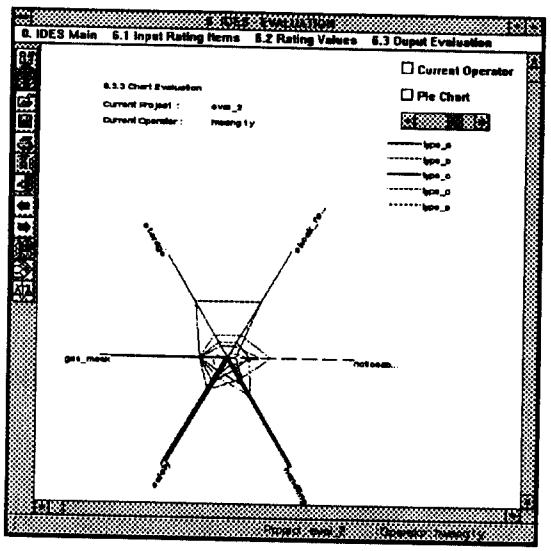
(Fig.4) Relative Value of Elements



(Fig.5) Superiority of Concepts



(Fig.6) Evaluation of Concepts

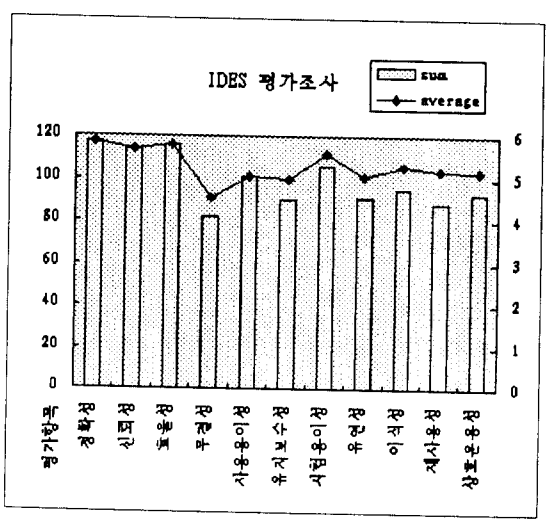


(Fig.7) Accumulative Evaluation

4) 테스트

프로그램 동작이 사양을 만족하고 있는지 확인하는 공정이다. 특히 시스템 테스트를 사용자 주도로 실제의 데이터를 사용하여 시스템 전체의 안정성이나 신뢰성을 테스트한다.

IDES 품질의 평가는 주어진 요구를 만족시키는 제품 혹은 서비스의 질을 보존하는데 필요한 것으로, 본 시스템의 평가를 위하여 산업 디자인 분야 디자이너, 교수 등으로 구성된 1차 평가에서 정확성(correctiveness), 신뢰성(reliability), 효율성(efficiency), 무결성(integrity), 사용용이성(usability), 유지보수성(maintainability), 시험용이성(testability), 유연성(flexibility), 이식성(portability), 재사용성(reusability), 그리고 상호운용성(interoperability)의 항목을 7점 SD척도로 측정하였다. 그 결과 상당히 만족스러운 긍정적 결과를 보이고 있다. 특히 정확성, 신뢰성, 효율성에 높은 반응을 보인데 비해 무결성, 유지보수성, 유연성 등에는 그다지 만족하지 못하고 있음을 나타내고 있다.



(Fig.8) Survey of IDES System Evaluation

또한 주관식 조사에서 매뉴얼, 처리순서 자동이동, 한글화 그리고 이미지 조작 등 사용상의 문제점을 후속 연구에서 보완할 필요가 있다. [12] (Fig.8)

3. 결 론

디자인은 필요에서 시작하여 그 필요를 충족시키는 제품의 출현으로 완성되는 목적 지향적 문제해결 활동(A Goal-Directed Problem-Solving Activity)인 것이다. 이를 통하여 인간의 삶의 질이 향상되고 궁극적으로는 보다 높은 삶의 가치를 구현할 수 있게 되는 것이다. 이와 같은 디자인 현상 속에서 다차원적 속성을 지니는 디자인 대상에 대한 통합적 방법의 적용으로 성공적인 신제품 생성이 가능할 것이라고 본다. 1차 연구에서는 디자인 평가 시스템의 객체지향 모델 설정에 중심을 두어 기초 연구를 진행한 바 있으며, 본 연구인 2차연구에서는 보다 구체적인 총합적 평가 시스템의 시작품(Prototype)의 개발에 객체 지향적 GUI의 적용하여 연구/개발하였으며, 금후 3차 연구에서는 멀티미디어 DB의 연계와 실용적인 평가시스템의 개발을 계획하고 있다.

앞에서 논의한 디자인의 통합적 평가시스템 특성은 다음과 같이 요약된다. 첫째로 최선의 해결안을 효과적으로 산출하기 위한 적절한 접근으로 다속성 평가항목에 따른 해결안의 평가를 수량화하며, 둘째로 이와 같은 평정작업을 컴퓨터화하여 신속.정확하게 처리하여 실무에 효과적이며, 셋째로 다수(경영자, 엔지니어, 마케터, 디자이너 그리고 소비자)의 집단적 평가 사고과정의 통합처리와 이를 위한 판단을 간편화시켰으며, 넷째로 평가과정이 이미지 DB구축을 바탕으로 시각화됨으로서 각종 평가 리포트는 뛰어난 전달성 및 공식화의 가능성을 지니므로 그 객관적 설득력을 높이게 될 것이다.

디자인 프로세스 위에서 디자이너들은 불가피하게 단독으로 또는 협동적으로 의사결정을 해야만 하며, 이것은 또한 최고 경영자(Top Management)의 의사결정에 영향을 미치게 되고 궁극적으로는 제품의 가치를 좌우하게 되는 것이다. 특히 디자인 평가에 있어서 디자인 목적과 상호 관계, 관련된 가치의 진가를 인정하고 측정하고 측정하는 것, 그리고 그 가치들 상호관계를 포함한 적절한 평가방법의 적용에 의해 디자인 최적화에의 접근이 가능하게 될 것이다. 특히 IDES는 디자인 평가의 효율성을 높여 준다. 다속성을 지니는 복잡한 평가작업을 다속성 평가 알고리즘을 적용하여 시간적 지체를 제거하였다. 따라서 의사결정자들은 각각의 경우에 따라 오직 판단만을 함으로

서 평가는 성력화되며, 프로젝트 소요기간은 단축되는 것이다.

그러나 다른 한편으로 제품 디자인을 보다 본질적으로 인간의 문제해결에 접근시키는 과정으로서 본다면, 이와 같은 평가 시스템의 맹목적 적용보다는 보다 원천적인 문제의 본질에 대한 디자이너의 인식을 전제로 인간생활을 만족시키는, 특히 시장/사용자 요구사항(Market/User Needs and Demands)을 충족시키는 도구로 자리잡아야 할 것이다.

그러나 보다 완전한 시스템 효용을 발휘하기 위한 IDES의 금후의 연구과제는 첫째로 CAD, 3DS 등의 어리케이션 시스템(Application System)과 데이터의 직접적인 호환이 요구되며, 둘째로 일반적 멀티미디어(Multi-media)의 DB와 자료의 호환이 연구되어지고, 셋째로 IDES 시스템 내부의 문제로 자동처리순서, 한글화 및 매뉴얼, 인쇄처리 등의 문제를 해결하여 사용자 중심의 시스템으로 발전시켜야 할 것이다.

FOOTNOTES

1) L.Bruce Archer, *System Method for Designers* in Developments in Design Methodology (Chister:John Wiley & Sons,1984),p60.

2) Bryan Lawson, *How Designers Thinks* (London:The Architectural Press,1980), p43.

3) Steven J. Fenves, *A Methodology for the Evaluation of Designers for Conformance with Standards*, Vol. 4 of Design Policy (London:The Design Council,1984),pp41-48.

4) S.A. Gregory, *Education:A Prelude*, Vol. 4 of Design Policy(London:The Design Council,1984),pp6-11.

5) 禹興龍,「定性的 디자인 價値의 數量化 硏究」(博士學位 論文, 漢陽大學校, 1990),pp 82-92.

6) Stuart Pugh, *Total Design* (Wokingham,England:Addison-Wesley Publishing,1991), p4.

7) Grady Booch, *Object-Oriented Design With Applications* (Benjamin/Cummings Publishing, 1991), pp25-132.

8) 송영재,「소프트웨어 엔지니어링」(홍릉과학, 1991), p41.

9) 이주헌,「소프트웨어 공학론」(법영사, 1993),p174.

10) 송영재, op.cit., p151.

11) 이주헌, op.cit.,p219.

12) ibid., p496.

BIBLIOGRAPHY

· Booch, Grady. *Object - Oriented Design With Applications.* Benjamin/Cummings Publishing, 1991.

· Coyne, R. D.;Rosenman, M. A.;Radford, A. D.; Balachandran, A.D.; and Gero, J.S. *Design System.* Addison-Wesley, 1990.

· Jacobson,Ivar; Christern,Magnus; Jonsson, Patrik; and Overgaard, Gunner. *Objected-Oriented Software Engineering.* Addison-Wesley, 1992.

· Jones,Greogory W. *Software Engineering.* John Wiley & Sons, 1990.

· Kirwan, B. and Ainsworht, L. K. eds. *A Guide to Task Analysis.* Taylor & Francis, 1992.

· Lafore, Robert. *LAFORE'S WINNDOWS PROGRAMMING MADE EASY.* Waite Group Press,1993.

· Lewis, T. G. *CASE-Computer-Aided Software Engineering.* New York:Van-Nostrand Reinhold, 1991.

· Mackay, Jeff. *Windows Programming with BORLAND C++.* McGraw Hill,1993.

· Pinson, Lewis J. and Weiner, Richard S. eds. *Application of Object-Oriented Programming.* Addison-Wesley, 1990.

· Roy, Robin and Wield, David eds. *Product Design and Innovation.* Open University Press, 1989.

· Rouse, William B. *Design for Success.* Wiley - Interscience, 1991.

· Scheuing, Eberhard E. *New Product Management.* Merril, 1989.

· Winblad, Ann L.; Edwards, Samuel D.; and King, David R. *Object-Oriented Software.* Addison-Wesley, 1990.

부 록

Class of Paired Comparison

```
. . .
class Pared_Comparison : public TDialog
{
  private:
    PTStatic text1;
    PTStatic text2;
    int count1, count2;
    char s1[MAXSTRLEN], s2[MAXSTRLEN];
  public:
    Pared_Comparison::Pared_Comparison( PTWindowsObject A
Parent, LPSTR AName );
    void great( RTMessage Msg ) = [ID_FIRST + 101];
    void some( RTMessage Msg ) = [ID_FIRST + 102];
```

```
    void not( RTMessage Msg ) = [ID_FIRST + 103];
    void cancel( RTMessage Msg ) = [ID_FIRST + 104];
    void nextstep( void );
};
void Pared_Comparison::nextstep( void )
{
  if( count2++ >= nElementNum-1 ) count1++, count2 = count1+
1;
  if( count1 >= nElementNum-1 ) CloseWindow( 0 );
  strcpy( s1, szElementNames[count1] );
  strcpy( s2, szElementNames[count2] );
  text1->SetText( s1 );
  text2->SetText( s2 );
}
Pared_Comparison::Pared_Comparison( PTWindowsObject APare
nt, LPSTR AName )
                          : TDialog( AParent, AName )
{
  int i, j;
  count1 = 0;
  count2 = 1;
  strcpy( s1, szElementNames[count1] );
  strcpy( s2, szElementNames[count2] );
  text1 = new TStatic( this, -1, s1, 20, 22, 100, 20, MAXSTRLE
N );
  text2 = new TStatic( this, -1, s2, 20, 70, 100, 20, MAXSTRLE
N );
};
void Pared_Comparison::great( RTMessage Msg )
{
  nCompareData[count1][count2] = 2;
  nextstep();
}
void Pared_Comparison::some( RTMessage Msg )
{
  nCompareData[count1][count2] = 1;
  nextstep();
}
void Pared_Comparison::not( RTMessage Msg )
{
  nCompareData[count1][count2] = 0;
  nextstep();
}
. . .
```