

공동협력 관계를 통한 GUI디자인 방법적 고찰

- A Review of Graphical User Interface Design Methods through Collaborative Teamwork

황지연 : LG 전자 디자인 연구소

서정운 : LG 전자 디자인 연구소

- 1. 서론
- 2. 공동협력(Collaboration)관계를 통한 GUI디자인 방법(Methods)
 - 2-1. 시각언어전개 방법
 - 3-1-1. 문제
 - 3-1-2. 원리
 - 3-1-3. 시각언어 적용
 - 2-2. Specification 시스템(세부명세 시스템)
 - 3-2-1. 문제
 - 3-2-2. 기본전략서
 - 3-2-3. 네비게이션 시나리오(Navigation Scenario)와 리소스 분류도 (Resource Library)
 - 3-2-4. 파일 네이밍 구조(File Naming Structure)
 - 2-3. 사용성테스트
 - 2-3-1. 문제
 - 2-3-2. 프로토타이핑(Prototyping)
 - 2-4. 디자인 평가
 - 2-4-1. 문제
 - 2-4-2. 평가기준
- 3. 결론
 - 참고문헌

(요약)

뉴미디어의 출현과 디지털 영역의 확장으로 인터페이스에 관한 디자이너의 역할이 점점 늘어나고 있지만 아직 디자인 커뮤니케이션에서는 아직 인터페이스 디자인의 기본 개념 및 방법론이 정립되지 않고 있다. 중요한 이유 중의 하나는 기존의 그래픽디자인이 일방적 커뮤니케이션이었다면 인터페이스 디자인은 쌍방향 커뮤니케이션, 즉 인터랙션이라는 것이다.

본 논문은 GUI 디자인의 사용성을 강조하기위해 소프트웨어 산업에서 일어나는 주요 문제와 디자이너의 의식 구조 문제를 논하고 이를 해결하기위한 방법을 제시하였다. 그 방법이란 소프트웨어 개발에 있어서, 공동협력(Collaboration) 모형에 의거한 시각 디자인의 구체적인 내용인데, GUI 디자인의 기본적인 시각언어 전개 방법과 원리, 시각언어를 객관화하기위한 Specification 시스템, 사용성 테스트에서의 문제와 프로토타입의 중요성, 마지막으로 디자인 평가 방법등의 문제등을 설명한다.

결론적으로 GUI 디자인의 사용성을 위해서는 디자이너가 다루

는 시각언어의 논리적 체계성 및 인지적 접근을 강조하였는데, 이는 디자이너와 공동 협력을하는 프로그램 엔지니어 및 제품 개발 구성원들과 커뮤니케이션하는데 있어 객관적 수단이 된다.

(abstract)

In the "digitalized" world of new media, visual designers are increasingly enlarging their roles in designing interaction products. However, the concepts and their methods of the interaction design have not been well established in the design community yet. The reason is attributed to the fact that designing interaction products requires two way communication between the products and final users. whereas traditional graphic design counts only on one way communication.

This study emphasizes GUI design should aim at improving the usability of products which helps users quickly learn about the products and easily use them. The usability of GUI design significantly improved by securing logical rationality helps designers communicate and collaborate with program engineers and other product developers. To explain how designers can effectively communicate with program engineers and other product participants in the design process, a model of design collaboration is provided in this study. Based on the model, the study deals with the principles and methods of visual interaction, the formulation of specification system, testing design usability, and design evaluation.

The study concludes that GUI designers must ensure logical rationality and the cognitive approach of visual language. Because it improves the usability of products. It also becomes an objective tool for communicating and collaborating with engineers and the other participants in the design process.

Key words : Usability, collaboration, methods

1. 서론

세계적인 디자인 전문자인 커뮤니케이션 아트(Communication Arts)가 주관한 제 1회 인터랙티브 디자인 공모전의 심사 내용이 '95 연감에 기사로 실린적이 있다. 여기에서는 내비게이션 구조(Navigation Structure: 사용자의 정보를 유도하는 흐름), 내용의 조직화, 논리적 방법, 사용상의 적절한 도움말 지원 등과 같은 사용성(Usability)에 관한 항목들이 주요 심사 기준으로 제시되어 있다. 이는 인터랙션디자이너가 더이상 독창성과 창의성의 표현예술이 아니고 인터랙션 디자인을 시각화(Visualization)하는데 있어 사용성이 핵심적인 이슈임을 말해주고 있다. 즉, 사용자의 니즈(Needs)에 도달하기위해 발생하는 문제해결의 논리적 체계이며 인터랙션 과학임을 강조하고 있는 것이다.¹⁾

GUI (Graphical User Interface: 그래픽을 이용한 사용자와의 대화방식) 디자인의 사용성은 네트워크 컴퓨터의 등장으로 중요성이 더욱 부각될 것이다. 네트워크 컴퓨터 시스템에서는 사용자가 서버에 내장되어 있는 수많은 소프트웨어의 브라우징(Browsing)을 통해 자신이 필요한 기능의 소프트웨어를 선택한다. 이때 소프트웨어의 기능을 한눈에 파악할 수 있는 인터페이스 디자인이 소프트웨어 제품의 성공에 결정적 요인이 될 것이다. 또한 개개의 소프트웨어 제품보다는 여러개의 제품에 공통으로 적용될 수 있는 인터페이스 언어와 표준을 개발하는 것은 소프트웨어를 생산하는 기업의 과제이다. 그렇다면 효과적인 커뮤니케이션에 성공하는 인터페이스 디자인을 위해 현재 어떠한 문제가 존재하는지 소프트웨어 제품개발 프로세스와 디자이너의 의식구조의 두가지 측면에서 간단히 살펴보자.

첫째, 소프트웨어 제품개발의 프로세스에서의 문제점을 살펴보자. 첨단 제품의 비즈니스를 둘러싸고 있는 마켓 환경은 결코 단순하지 않다. 혁신적인 제품이 항상 성공하는것은 아니며 훌륭한 제품을 적절치 않은 시기에 출시해서 실패한 사례는 얼마든지 있다. 성공적인 기업조차도 새로운 제품세대로의 전환을 시도하기 위해 어려움을 겪고 있다. 예를 들면 만약 소비자가 새로운 제품이 출시될 것이라는 정보를 갖게 된다면 기존 제품의 매출이 떨어지고 새제품의 성능이 증명될 때까지는 새제품은 팔리지 않을 것이다. 이같이 첨단 소프트웨어 제품 개발에 있어 일정문제는 성공과 실패에 상당한 민감성을 갖는다. 왜냐하면 소프트웨어 시장에서 누가 먼저 소비자에게 사용자 인터페이스를 제공하여 그 제품의 대중적 친근함을 얻느냐가 성공의 중요한 요인이 되기 때문이다. 우수한 소프트웨어를 개발하기 위해서는 개발 기간보다 테스트나 디버깅(

Debugging)에 더 많은 기간이 소요되는데, 우리나라의 많은 기업들이 생산 일정때문에 제대로 테스트나 디버깅하지 않은 소프트웨어들을 출시하고 있다. 그 이유는 아직 우리 기업이 소프트웨어 개발의 초기 단계에 제품을 기획할 수 있는 능력이 부족하여 개발 후기에 빈번히 스펙(Specification)변경을 하기 때문이다. 따라서 보통 테스트나 디버깅에는 투자할 시간이 없는데는 것이다. 소프트웨어 제품 기획에서 가장 문제가 되는 것은 결국 사용자 인터페이스에 관한 의사결정인데 사용자 인터페이스에 관한 개개인의 주관적 견해(Flexibility)는 의사결정을 어렵게 하는 결정적인 요인이다. 예를 들면 하드웨어의 기능스펙 결정은 기술적 한계와 비용의 문제로서 비교적 명확한 의사결정이 용이한데 비해 소프트웨어 사용자 인터페이스 기능 설계는 개인의 관점 차이로 많은 의견 차이가 있기 때문이다. 결국 이러한 환경은 프로그래밍뿐만 아니라 디자인의 통합적 기획을 어렵게 하는 주요 요인이 된다.

둘째, GUI에 대한 디자이너의 이해 부족 문제를 들 수 있다. 뉴 미디어의 등장 이후 시각 커뮤니케이션 디자인영역에 인터랙션 디자인 개념이 새로 추가되었다. 그러나 디자인 커뮤니티안에서 이러한 개념들을 정리하고 하나의 학문적 영역으로 개발하기에는 아직 시간이 필요하다. 우리가 지금 자주 대하는 컴퓨터 스크린에 떠있는 그림 조각들, 즉 아이콘, 버튼 등의 그래픽들이 컴퓨터와 사용자의 대화를 유도하는 주요 도구로서 대중화된 시기는 그리 오래되지 않았기 때문이다. 1981년 제록스스타(Zerox Star)가 처음으로 GUI를 사용자 인터페이스로 적용했으나 GUI를 대중화하는데 역할을 한 것은 1984년 맥킨토시의 애플리사(Apple Lisa)²⁾였고 주로 IBM PC가 일반 사용자의 대부분인 우리나라에서 GUI가 대중화된 것은 1991년 마이크로소프트사의 윈도우 3.1을 통해서 일 것이다. 사람과 컴퓨터의 인터페이스에서 시각적으로 그래픽이 이용된 것은 무엇보다도 사용하기 쉽다는것이 이유이며 그것이 인터페이스 개발의 존재 이유이다. 그런데 아직도 많은 디자이너들이 GUI의 표준을 이해하지 못하고 있으며 인터페이스에서 사용성보다는 감각적 스타일에 치중하는 경향을 보인다. 기존의 디자이너가 인터페이스 디자인의 맥락을 이해하고 목적에 맞는 디자인 프로세스와 방법을 갖기에는 10년이란 기간은 짧다고 할 수도 있다. 그러나 소프트웨어가 디자인으로 경쟁력을 갖춰야 하는 시기는 이미 성숙했다. 소프트웨어는 일반에게 더이상 신기한 기능의 제품이 아니기 때문이다.

스타일보다는 커뮤니케이션을 추구하며, 시각적 컴포넌트의 트렌디한 스타일보다는 시각적 컴포넌트들이 기능과 연결되어 전체를 이루는 유기적 관계를 추구하는 GUI디자인에 관

1. 본 논문에서는 인터랙션 디자인의 용어를 개념적으로 가장 범위가 넓게 사용하고 소프트웨어의 인터랙션 디자인의 설계에 관한 구체적 내용을 지칭할 때는 인터페이스의 용어를 사용하였다.

2. Aaron Marcus, Graphic Design for Electric Documents and User Interface, 1992 P 139

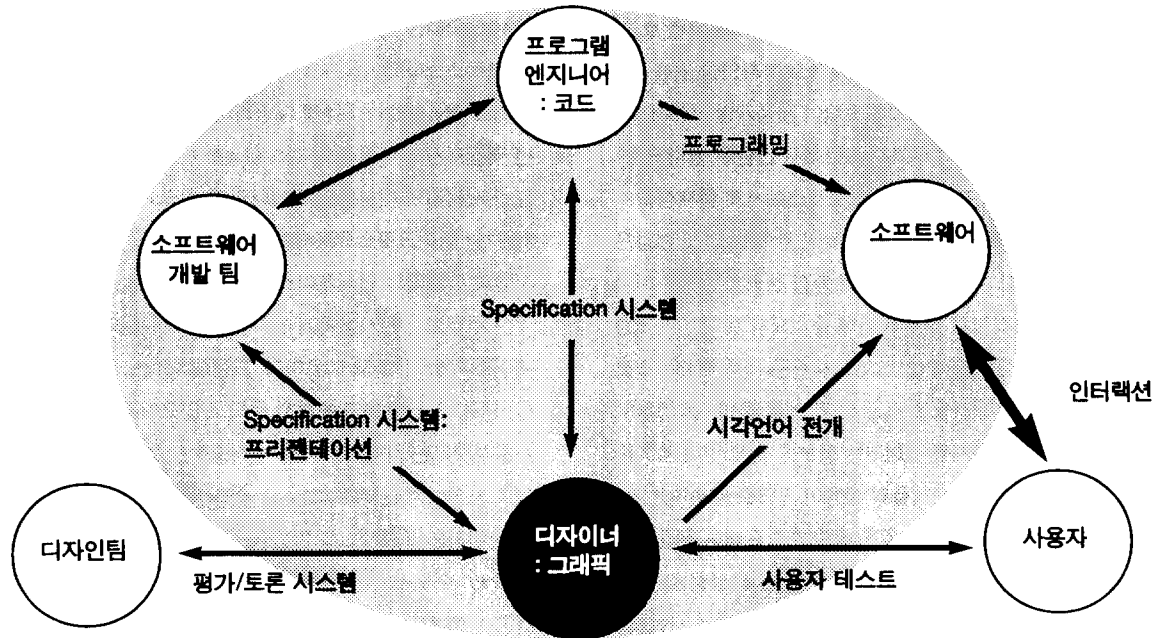
한 방법론은 GUI디자인 연구의 시작이다. 디자인 방법론은 표출된 논리와 사고의 과정을 설명한다. 그것은 디자이너가 무엇을 해야하고 왜 해야하는가에 대한 지식이다. 방법론은 보통 디자인 프로세스안에서 설명하는데 그동안 학자나 디자이너에 의해 다양한 관점의 모델들이 개발되어 왔다. GUI디자인의 디자인 프로세스는 기존의 모형과 같은 분석과 종합이라는 기본 개념안에 존재한다. 따라서 본 논문에서는 새로운 디자인 프로세스의 모형을 제시하는것이 아니라 인터페이스 분야의 특이한 공동협력의 커뮤니케이션 모형안에서 필요한 방법론을 제시하고자 한다.

3. 공동협력(Collaboration)관계를 통한 GUI디자인 방법 (Methods)

GUI를 공간으로하는 소프트웨어는 코드와 그래픽으로 이루어진다. 즉, 프로그램 엔지니어와 디자이너가 소프트웨어 개발의 기본 구성원이라 할 수 있다. 이 두 그룹 이외에 소프트웨어개발 팀에는 서로 다른 문화적 배경이 다른 그룹들이 참여한다. 만약 음악 편집 소프트웨어를 만든다면 음악 전문가가 참여할 것이고 인터페이스 설계에는 인지심리학적 지식이 기반이 될 것이다. 그래서 소프트웨어 인터페이스 분야에는 이러한

다양한 학문들(Multidisciplinary)이 사람과 컴퓨터의 효과적인 인터랙션이라는 목표를 갖고 학제간 연구(Interdisciplinary)를 하는 것이다. 전문적 GUI 디자이너라면 혼자서 우수한 제품을 만들 수 있다는 생각보다는 개발 팀 모두의 공동협력을 통해 디자인이 이루어져야 한다는 인식이 전제되어야 할 것이다. 디자인이 아무리 조형적으로 뛰어나도 프로그램 엔지니어와의 커뮤니케이션에 문제가 생기면 디자이너가 의도한 디자인이 나올 수 없기 때문이다. 또한 디자인 과정에서도 디자이너의 조형적 관점 이외에 프로그램의 효율성, 마케팅, 사용성 등의 많은 문제들 속에서 올바른 의사결정이 이루어져야 한다.

다음의 그림은 디자이너를 중심으로해서 소프트웨어 개발을 둘러싸고 있는 그룹과 어떤 커뮤니케이션과 과제가 있는지 보여주고 있다. 가장 기본이 되는 시각언어 전개방법, 프로그래머와의 커뮤니케이션을 위한 Specification 시스템, 사용성의 원성도를 위한 사용성 테스트와 평가 등, GUI디자인에 있어 3가지 주요 이슈들을 공동협력이라는 맥락안에서 살펴 보았다.



▶ GUI 디자인 공동 협력 모형

3-1. GUI시각언어전개 방법

3-1-1. 문제(Problems)

소프트웨어의 프로그램 코드는 75%이상이 사용자 인터페이스에 사용된다. 3) 따라서 GUI디자인은 새로운 이미지를 창출한 다기보다는 정보의분석을 통한 시각언어의 프로그래밍이라고 할 수 있다. 이 말은 GUI디자인이 심미성을 무시해도 된다는 의미는 아니다. 다만 GUI의 심미성은 예술적 관점에서의 독창성이나 특이적 조형미의 시험장이 되기 보다는 커뮤니케이션 관점에서의 조화와 통일이다. 그리고 현재 의 소프트웨어 제품의 적합성에 기초한 심미성이어야 한다는 것이다. 그러나 현실적으로 심미성의 문제는 단순하고 명확하지는 않다. 자주 디자이너들이 트렌디한 스타일 개발에 심취하는데 이는 디자이너가 사용성의 문제해결, 그 자체 보다는 그 이상의 시각적 표현을 꿈꾸는 특성을 가지고 있기 때문이다.4) 그래서 디자이너가 소프트웨어의 시각적 표현 스타일에 고민하는 실제적인 문제는 트렌디한 스타일과 전통적스타일 간의 선택이 될때가 많다. 그런데 이런 경우 인터페이스 디자인 이론은 새로운 스타일에 확신이 부족하면 기존의 방식을 택하라는 것이다. 실제로 전통적 스타일의 디자인이 트렌디한 디자인보다는 효과적인 커뮤니케이션을 이끄는 경우가 많다고 전문 인터페이스 디자이너들은 말한다.5) 이러한 예는 인터랙션 디자인의 시각적 디스플레이가 의도된 목적과 분리되었을때 그 시각적 디스플레이가 가진 심미성에 의미가 없어진다는 것을 말해 준다. 많은 경우 GUI디자인의 프로세스는 이러한 트레이드오프 (Trade-Off) 6) 의 문제를 해결해 나가는 과정이라고 할 수 있다. 트레이드오프란 두가지의 목적중에서 하나를 포기해야하는 상황에서의 결정을 말하는데, 디자이너에 있어서 자주 발생하는 트레이드오프는 심미적 측면과 사용성측면이다. 즉 조형적으로는 아름다우나 일관성의 커뮤니케이션의 원리에 위배되는 경우가 종종 발생한다. 이때 디자이너가 사용성 즉 커뮤니케이션 원리보다는 심미적 아이디어에 심취되어 있기 때문에 현명한 트레이드오프를 하지 못하는 경우가 있다. 물론 전문적 디자이너라면 제품의 상황에 따라 두가지의 상충되는 개념의 어느 쪽에 무게를 실어야만 하는가를 알고 있을 것이다. 예를 들면 오피스용 소프트웨어와 어린이용 소프트웨어가 추구하는 목적은 서로 다를것이다. 또한 윈도우 표준을 따르는 소프트웨어와 윈도우를 기본으로 하되 약간의 비주얼 아이덴티티를 갖는 소프트웨어에서의 시각언어는 다를 것이다. 너무 많은 트레이드오프 상황을 피하고 효율적인 디자인 개발에 필요한 것은 인터랙션 원리에 입각한 방법이

다

3-1-2. 기본원리

최근에는 첨단 상품 디자인 개발에 있어 사용자 중심의 연구는 그 어느 때보다도 강조되고 있다. 이러한 사용자 연구를 바탕으로 상품의 컨셉이 결정되면 적합한 시각 디자인이 전개되어야 한다. GUI디자인에 있어 시각 디자인 기획은 일련의 시각적인 Rule을 체계화하는 것이다. 소프트웨어의 기능과 정보를 나타내는 이미지들을 통합, 조직하는데는 많은 용어가 있으나 단순성(Simplicity), 일관성(Consistency), 명료성(Clearity), 효율성(Efficiency)은 사용자와의 커뮤니케이션 관점에서 시각적 Rule을 체계화하는데 가장 적합한 기본 원리(Principle)가 되고 있다. 이러한 원리를 중심으로 GUI 시각언어를 전개해 나가는 구체적 방법을 제시하기는 쉽지 않지만 다음에서 그 방향을 제시해 보려고 한다.

3-1-3. 시각언어 전개의 방법

디자인해야할 대단위의 정보를 시각화하는데 우선 단순성의 원리를 생각해 보자. 여기에서의 단순성은 복잡성(Complexity)의 반대어미인 적은 정보의 양을 말하는 것이 아니고 정보의 통합화(integration), 조직화를 말하는 것이다. 정보를 단순하게 하기위한 유기적 통합화의 방법은 전체의 정보를 중요도, 성격등의 몇가지 패러미터(Parameter)로 분류하는데서 시작한다. 분류에서 선택적그룹핑의 단계로 나타난다. 효과적인 그룹핑이론은 게스탈트이론이 기본이 되는데, 그것은 사람이 어떤 정보나 시각물을 인지할 때 한번에 덩어리가 되어 파악되는 원리를 4가지 기본 용어로 설명한 것이다. 근접성(Proximity), 유사성(Similarity), 계속성(Continuity), 닫힘(Closure)의 기본 법칙 7)은 커뮤니케이션을 위한 그룹핑에 있어 중요한 원리가 된다. 다시말하면 그룹핑은 많은 정보가 하나씩의 덩어리가 되어 우리의 인지에 단순한 Map을 형성하는 역할을 한다. 따라서 정보의 단순성을 얻기 위해서는 효과적인 그룹핑이 주요 방법이 된다는 것이다. 그렇다면 이러한 그룹핑은 어떤 방법으로 하는가? 그 그룹핑 방법이 바로 Color나 형태, 텍스춰,레이아웃등의 시각언어 요소들로 이루어진다고 할 수 있다. 아래의 예는 사용자와의 인터랙션에서 시각언어의 요소들이 어떻게 이용될 수 있는지 구체적으로 보여 준다. 예를 들면 정보의 계층을 수직적 내비게이션안에서는 레이아웃의 변화로 나타낼 수 있고 수평적 그룹핑은 색상으로 나타낼 수 있다. 이러한 그룹의 정보들이 전개될 때에는 일관성이 사용자에게 각 정보의 기능을 이해하고 기억하게 한다. 이러한 일관성으로 이루어진 그룹들과의 관계는 일련의

3. Terry Winograd, Bringing Design to Software, 1996, P. 5.

4. Terry Winograd, Bringing Design to Software, 1996, P. 151

5. Virginia Howlet, Visual Interface Design for Windows, 1996, P. 41- 43.

6. 윤철호, 인간, 컴퓨터 인터페이스, 1996 P. 44.

7. D. W. Hamlyn, The Psychology of Perception P. 67

Rule로써 명시될 수 있어야 한다. 예를 들면 도움말은 항상 오른쪽 상단에 위치한다거나, 전체 정보의 관리에 관한 기능들의 그룹은 파랑계통의 Color로 나타낸다거나, 각 버튼간의 간격은 5 픽셀이어야 해야하는 일련의 Rule이다. 따라서 모든 시각 요소들은 사용자의 기억에 코딩됨을 전제로

디자이너의 시각 이미지에 관한 창의력과 스킬은 각 이미지 요소를 만들어내기위해 활용되는데 비해 논리적 합리성은 이러한 시각언어를 이해하고 통합을 향해 각 요소들을 조율해 나가는 과정에서 활용되며 이 모드는 GUI에 있어 효과적 인터랙션을 목표로하는 바이다.

다음의 예는 GUI의 오브젝트 메뉴의 포커싱(Focusing)과 선택(Selecting)의 인터랙션 방법을 시각 언어를 이용할때 어떤 예가 있을 수 있는지 보여준다.

1. 반전 : 오브젝트의 이미지가 반전됨으로써 포커싱이 표현된다.
2. 외곽선(Outline) : 오브젝트의 이미지 주위에 외곽선이 그려짐으로써 포커싱이 표현된다.
3. 명암효과 : 오브젝트의 명암이 변하거나 오브젝트 주위의 명암이 스포트라이트의 형태로 표현될 수 있다.
4. 채도효과 : 오브젝트가 무채색이었다가 고채도로 변함으로써 포커싱이 표현될 수 있다.
5. 동영상 효과(Animation) : 오브젝트의 특성을 표현하거나 형태를 왜곡시키는 표현을 이용할 수 있다. 그러나 동영상 효과는 반응의 정도가 크므로 많은 메뉴나 자주 쓰는 기능에서 이용하면 사용성이 오히려 떨어질 수 있는 면을 고려해야한다.
6. 입체효과 : 오브젝트 버튼으로 표현되거나 평면에서 버튼으로 변할 수도 있고 그림자를 줄 수도 있다.
7. 위치 이동 : 위치가 변함으로써 포커싱이나 선택을 표현할 수 있다.
8. 커져 변화 : 오브젝트를 가르키는 커저에 효과를 줄 수 있다. 위와 같은 시각언어들의 요소는 소프트웨어의 내비게이션 구조상에서 그룹핑에 적용하여 위에서 언급한 인터랙션의 기본 원리에 따라 사용되어야 한다.

3-2. Specification 시스템(세부 명세서 및 명세도)

3-2-1. 문제(Problems)

프로그래머와 디자이너에 의해 만들어지는 코드와 그래픽은 GUI 소프트웨어를 완성하기위한 기본 요소이다. 공동협력 학제간(Collaborative Interdisciplinary) 팀의 필요성이 강조된 이후로 이 두 그룹간의 이질적 사고 방식(Different Thinking Styles)은 HCI의 테마의 중의 하나가 되었다. 스코트 킴(Scott Kim)은 그의 논문에서 이러한 문제에 대해 다음과 같은 대화를 인용했다.

Designer: Can't you see what's in front of you?

Programmer: Can't you tell me the rules? 8)

여기서 의 문제는 디자이너는 시각적인 사실로 커뮤니케이션 할 수 있다고 생각하고 엔지니어는 시각적 사실에 관한 법칙이나 규정을 설명해 주길 요구하고 있다.

규모가 큰 소프트웨어 프로젝트에서는 수십개의 화면과 그에 관련된 그래픽의 조각 화일들이 천개를 넘는 경우가 있다. 이런 경우 디자이너가 디자인을 엔지니어에게 넘겨주는 단계에서 정확한 디자인 의도가 제대로 커뮤니케이션되지 않는 경우가 발생한다. 왜냐하면 엔지니어의 주된 관심은 프로그램이 만들어져 움직이는데 있기 때문에 시각언어의 전체 구조와 의미를 모르는 상태에서 시각언어를 자세히 읽을 수 없기 때문이다. 그러한 상황은 디자인이 디자이너의 의도대로 적용되지 않고 결국 뒤떨어진 디자인 결과를 낳는다. 여기서 건축에 있어서의 청사진과 같은 역할을 하는 비주얼 인터페이스 디자인의 Specification 시스템이 필요하다. 디자이너는 커뮤니케이션만을 위한 Specification 시스템을 제작하는 것은 전체적 일정에서 비효율적이라고 생각할 도 있다. 그러나 Specification 시스템은 디자인 프로세스의 마지막 단계에서 커뮤니케이션의 수단으로만 사용되는 것만이 아니라 분석과 종합의 단계에서 부터 전체 디자인 작업의 윤곽(Frame)으로써 이용된다. 이는 단순성을 목표로 하는 디자인을 완성하기 위한 수단이다. 즉, 디자인의 비논리적인 면을 추적하고 발견해나가는데 수단으로 이용되기 때문에 인터랙션 디자인 프로세스가 선형적인 단계보다는 입체적으로 서로 연관되어 반복되어야 하는가를 보여 준다. 또 이 시스템은 그대로 프리젠테이션 방법으로 효과적으로 이용될 수 있는데 그 이유는 디자인 의도를 시각적으로 보여줄 수 있기 때문이다. Specification 시스템은 기본전략서, 네비게이션 시나리오, 리소스분류도, 인터랙티브 프로토타입과 같은 항목을 갖는다. 인터랙티브 프로토타입은 사용성 테스트에서 설명하기로 한다.

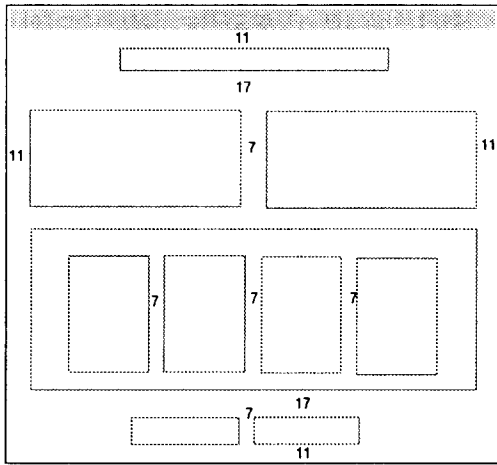
3-2-2. 기본전략서

이것은 효율성을 추구하기 위해 시각 표현을 단순화 함으로써 논리, 체계적인 시각적 규정(Rule)을 만드는 것이다. 내비게이션의 수직적, 수평적 관계를 시각적 언어로 규정하고, 그룹핑을 토대로 불필요한 시각 요소의 변화(Variation)를 줄여 수직적, 형태적으로 표현한 규정을 말한다. GUI디자인의 중요한 요소의 하나인 대화상자(Dialogue Box) 디자인을 예로 들어 보자. 대화 상자에 사용되는 기능은 대체적으로 명령버튼, 라디오버튼, 스크롤바, 타이틀 등으로 특정 타입이 몇가지 있다. 소프트

8. Brenda Laurel, The Art of Human Computer Interface Desin, 1990, 중에서 Scott Kim의 논문 Interdisciplinary Collaboration P. 31.

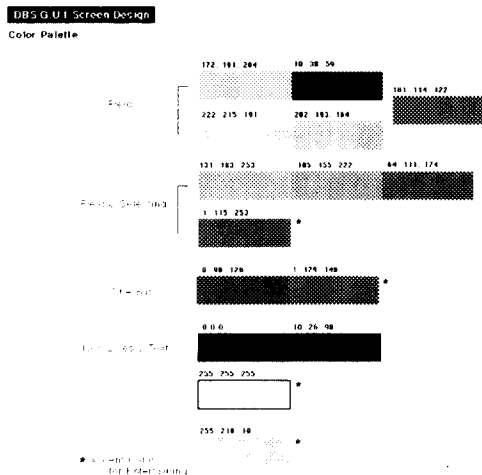
웨어마디의 대화상자를 일일이 디자인하는 것이 아니라 소프트웨어에서 나올 수 있는 대화상자의 경우를 추출해서 매뉴얼을 만드는 것이다. 기능을 표현하는 오브젝트들의 크기와 간격들도 3~4가지의 표준 크기를 정하여 크기나 간격의 변화(Variation)을 최소의 수로 하는 것이다. 이같은 일은 아직 마이크로소프트의 윈도우용 프로그램이나 맥킨토시의 응용프로그램에도 버튼의 크기나 형태만 정해져 있을 뿐 일관성 있는 세부적 간격규정은 되어 있지 않은 상태이다.

또한 색상은 시각 언어 중 가장 강한 인지력을 갖고 있기 때문에 GUI디자인 전략의 중요한 언어가 된다. 이는 소프트웨어의 정



▶ 대화상자의 전략 규정의 예

GUI디자인에서 스크린의 면적을 효율적으로 사용하는것은 기본사항이다. 대화상자안에서 상자와 상자간의 그룹핑을 위해 레이아웃상의 간격을 정하는 것은 게스탈트 이론의 근접성이론인데 보통 인지될 수 있는 간격들의 차이를 4가지로 정하면 일관성있게 적용가능하다. 위의 그림에서는 그중 7, 11, 17이라는3가지의 간격규정이 적용되었다.



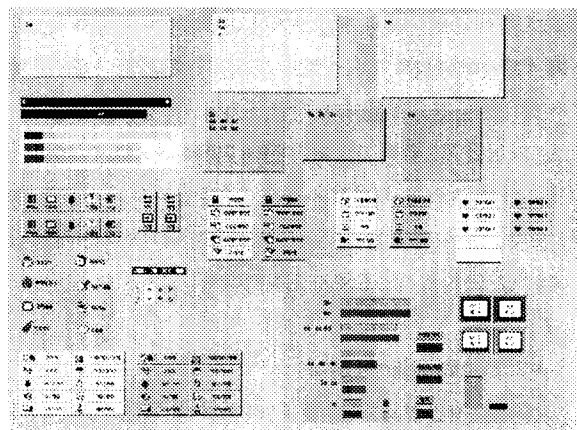
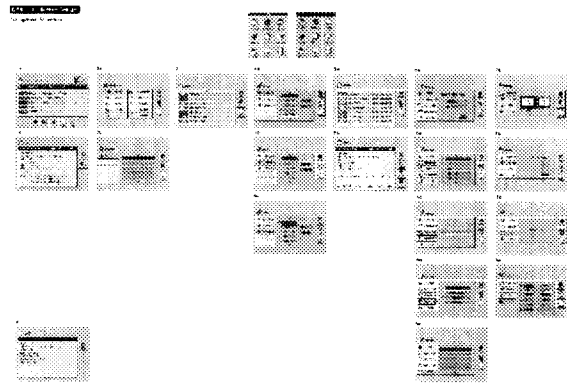
▶ 색상 팔레트의 예

특성을 가진 그룹간의 색상을 사용한 전략을 보여준다.

보나 기능을 색상으로 상징화하는, 일종의 색상의 코드화된 형태를 매뉴얼적인 규정으로써 보여주는 것이다. 방법은 소프트웨어 전체의 정보를 몇가지 중요 기준으로 분류하고 내비게이션의 수직적, 수평적 관계를 파악한 후에 색상이 어느 그룹핑의 도구가 될 것이가를 테스트해본다. Color의 사용은 색, 명도, 채도의 차이의 대비를 이용해 색상팔레트를 만들고 기능에 따른 역할을 명시하는 것이다.

3-2-3. 네비게이션 시나리오(Navigation Senario)와 리소스 분류도 (Resource Library)

네비게이션 시나리오는 리소스 라이브러리시스템과 함께 마치 건축의 설계도면과 같은 역할을 하는데 표준화된 크기와 형태의 그래픽 리소스들이 전체 설계도면에서 어떻게 쓰였는지를 한 눈에 보여주는 역할을 한다. 즉, 리소스 라이브러리에서는 장난감 레고블럭의 크기와 형태가 표준화 된 것과 같이



▶ 네비게이션 시나리오 및 리소스분류도

전체 구조에서 그래픽리소스들이 어떻게 사용되었나를 한눈에 보여주기 때문에 디자인의 비논리적 부분이나 문제점들을 쉽게 찾을 수 있다. 반복된 평가와 토론을 통해 논리적이고 효율적인 디자인을 목표로 하며 프로그램 엔지니어등 개발팀안에서 디자인의 전략 및 의도를 이해할 수 있는 커뮤니케이션의 도구이다.

그 스타일과 형태에 따라 단위화되어 전개도의 어느 부분에 쓰였는가를 명시한다. 또한 전체 리소스들의 변화(Variation) 수를 줄이는 단위화는 효율적 프로그래밍을 위한 디자인의 의도를 보여준다. 전체의 그래픽 메모리를 파악할 수 있는 자료로서 역할을 한다.

3-2-4. 파일 네이밍 구조(Naming Structure)

전체 소프트웨어 프로젝트에 사용되는 최종 그래픽 화일들만도 그 수가 300개가 넘는 일은 보통이다. 디자이너가 이러한 수의 화일들을 엔지니어에게 넘겨주는 과정에서 많은 혼동을 일으키며 작업의 비능률이 따른다. 더우기 그러한 화일들이 자주 수정되는 과정에서 혼란이 가중된다. 이에 화일의 이름을 정하는 데는 도서관 시스템과 흡사한 디자이너와 프로그램 엔지니어간의 약속이 필요하다. 예를 들면, 시각 요소의 그룹의 특성을 나타내는 기호, 각 화일의 특성을 나타내는 기호, 오브젝트의 선택 상태, 동영상 프레임, 수정 버전 등 각 소프트웨어의 특성에 따라 문자의 분류 시스템을 만들어 화일을 관리해야만 디자이너 자신은 물론 프로그램 엔지니어와의 효율적 커뮤니케이션을 유지할 수 있다.

3-3. 사용성 테스트

3-3-1. 문제

사용성 테스트의 가장 현실적인 문제는 생산일정이다. 사용성 테스트의 필요성과 가치는 개발자 누구나 공감하면서도 현실에 적용할 때 문제시되는 이유는 프로그램 코딩이 완성된 후에 시행되어야 하는 것이며 그 결과에 대한 수정작업은 프로그래머나 디자이너에게 막대한 업무를 초래할 수도 있기 때문이다. 그래서 최근에는 개발 초기에 이루어지는 사용자 테스트가 주목되며 개발후기 테스트의 결과는 그 제품의 다음버전에서 적용하는 것이 보통이다. 개발 초기의 사용성 테스트는 소프트웨어의 기능들이 코딩되기 전에 디자이너에 의한 하이피 프로토타입에 의해 부분적 사용성 이슈들을 나누어 실행할 수 있으나 이는 실제 사용자들에 의한 실험이 되기보다는 개발 팀 내부에서의 자체 테스트로서 유용하다.

3-3-2. 프로토타입(Prototype)

GUI 디자인 프로토타이핑(Prototyping: 모형제작)은 디자인 Quality를 높이는데 4가지 도구로써 중요하다.

- 1, 디자이너 자신이 GUI 디자인 기본원리를 적용하며 제작하는데 가장 훌륭한 디자인 도구가 된다.
- 2, 디자인 내부그룹에서의 Evaluation의 필수도구가 된다.
- 3, 엔지니어 및 타그룹과의 커뮤니케이션의 도구가 된다.

4, 초기 사용자 테스트의 도구가 된다.

이와같이 프로토타입의 방법개발과 그의 디자인 프로세스에의 적용은 GUI 디자인의 기본사항이다. 또한 효과적인 인터랙션을 위한 디자이너로서의 프로토타이핑은 두가지의 종류가 있다.

첫째는 로우피(Low Fidelity) 프로토타입으로써 불리우는데 실제 제품의 완성도에 못 미치지만 전체 구조나 흐름을 파악하는 도구로 사용되는 것을 말한다. 개발 초기에 이용되며 주로 종이나 스티커등 사용자가 제품을 짧은 시간에 직접 만들수 있도록 고안된 방법들이 개발되어 있다. 페이퍼 프로토타입은 가장 일반적인데 말 그대로 종이카드와 펜, 접착제, 마커등 전통적 그래픽 디자이너의 스케치 도구를 이용한 것이다. 한 사람의 아이디어를 표현하는 수단이 아니며 개발자 전체가 사용성테스트를 하면서 GUI 디자인을 위한 구체적 토론 방법이며 커뮤니케이션의 도구이다. GUI디자인이 완전한 윈도우 표준으로 설정되었을 때는 스크린 프로토타입없이 페이퍼 프로토타입만으로 프로토타입이 끝나는 경우가 많다. 이 방법의 잇점은 빠른 의견수렴과 결정으로 소프트웨어 개발의 일정문제 딜레마를 해결해주는 데 있다. 디자이너에게는 스크린 프로토타입을 위해 제작하는 화면 디자인의 프린트물로 이루어질 수 있다. 실제 크기의 화면들로 대형 보드에 네비게이션 시나리오를 만드는 방법인데 이미 Specification 시스템에서 설명한바 있다. 이 방법은 디자인 전체의 구조와 요소들의 관계를 한 눈에 파악해서 인터랙션 디자인 원리인 Simplicity, Consistency, Clarity를 적용하는 가장 유익한 도구가 된다.

두번째는 하이피(High Fidelity) 프로토타입으로 불리우는데 완성품에 가깝게 인터랙티브하게 만들어서 실제 사용자 테스트의 도구로 이용된다고 할 수 있다. 또한 제작과정에서 그래픽 인터랙션의 시뮬레이션 도구로써 훌륭한 역할을 한다. 완성도가 높아 이상적인 프로토타입이지만 상대적으로 시간이 투자되므로 디자이너 스스로 디자인 프로세스를 관리하는 Know-how가 필요하다.

스크린 프로토타입은 일반적으로 하이퍼카드(Hyper Card), 디렉터 (MacroMind Director), 비주얼 베이식 (Visual Basic) 이 대표적으로 비교된다. 이들은 모두 공통적으로 프로토타입을 쉽게 만들기 위한 도구이지만 아마추어에게는 하이퍼카드가 이용되고, 소프트웨어 엔지니어에게는 비주얼 베이식이, 디자이너에게는 세련된 시각적 효과를 위해 디렉터가 주로 이용된다.

3-4. 디자인 평가

3-4-1. 문제

디자인 평가는 사용성을 목표로하는 GUI디자인의 질을 높

9. Terry Winograd, Bringing Design to Software, 1996, P.191

이는데 중요한 부분임에도 활성화되기 어려운 몇가지 문제가 있어 왔다. 첫째로는 시각언어를 일반언어로 표현하는 테크닉에 익숙하지 못하고 시각언어의 다양한 측면과 주관성(Flexibility)을 객관화하는데 있어 어려움이 있다. 그러나 디자인 평가와 토론에 있어 가장 어려운 면은 디자이너 자신의 디자인 작업을 객관화하는 전문적 정신의 결여와 자기 확신의 부족이다. 이러한 상황에서 기준이 모호한 평가 내용과 항상 방어적 메카니즘이 작용하는 디자이너의 심리현상이 부딪쳐 이상적 형태의 디자인 평가를 어렵게 한다. 또한 전문적 GUI 디자인 평가 시스템을 위해서는 디자이너의 전문적 자세도 전제가 되어야겠지만 현실적이고 합리적인 평가 방법이 정착화 되어야 한다. 즉, 단순한 선호 스타일이 주제가 되어서는 안되겠으며 선택적 기준에 의해 진행되어야 하며 디자이너 그룹, 제품개발 팀, 사용자그룹 인터뷰등 다양한 평가팀 구성이 중요하다.

3-4-2. 디자인 평가기준

GUI디자인 프로젝트의 성격에 따라 평가의 검증 항목(Check list)은 디자이너 자신이 제시하여야 한다. 디자이너가 제시하여야 할 검증 항목은 소프트웨어 제품에 따라 약간의 내용이 다를 수 있는데 목적과 효과를 중심으로하여 작성되어야 한다. 다음과 같은 예를 들 수 있다.

- 전체적 디자인이 제품에 적합성
- 디자인 원리가 이용된 방법
- 시각요소들의 그룹핑의 합리성
- 그룹핑의 계층 표현
- 형태와 배경의 관계
- 각 그룹간의 명확성
- 타이포그래피의 가독성
- 색상 전략
- 프로그래밍의 효율적 측면

이 항목들은 소프트웨어의 성격에 따라 내용이 달라질 수 있으며 이 같은 명확한 주제들은 시각언어를 객관화하는데 기준을 마련한다. 또한 이러한 항목들을 디자이너 자신이 제시하려면 디자인 프로세스를 통해 시각언어를 사용하는 논리적 근거를 개인의 기호에 따른 스타일의 평가도 부분이 되어야 하겠지만 무엇보다도 정보를 나타내는 시각언어의 논리적 체계를 검증하는 과정이 되어야 한다.

4. 결론

디자이너의 두뇌 활동에는 두가지의 전형적인 모드가 있다. 첫 번째는 창의성(Creativity)이며 두 번째는 논리적 합리성(Rationality)이다. 창의성이란 가능성있는 대안들로부터 비합리

적 추정을 하는 과정에서 일어나며 논리적 합리성이란 논리적인 질서와 계산으로 문제해결을 하는 과정에서 일어난다. 크리스 존스의 용어로 표현하자면 창의력에 관련된 부분을 블랙박스(Black Box)라고 하며 생각의 과정이 겉으로 드러나는 논리나 체계에 관련된 부분을 글래스박스(Glass Box)¹⁰⁾라고 한다. 두가지 모드는 성격상 상반된 입장에 있으나 하나의 바운더리를 갖고 서로 관련지어(Interconnected)있어 그 정확한 경계를 긋기가 어렵다. 즉 하나의 디자인 프로젝트에서 어디까지가 창의력의 모드에서 해석이 되며 어디까지가 논리, 체계성의 모드에서 해석이 되는지 날카로운 구분이 어려운 것이다. 그동안 이 두가지 모드 중에서 창의성 즉 크리에이티비티는 디자인을 예술의 범주에 포함해서 설명하는 가장 중요한 요소로서 인식 되어 왔으며 디자인 커뮤니티안에서는 줄곧 논리적 합리성보다는 창의력을 강조해 왔다. 사실 이는 디자인을 일반 학문과 구분짓는 대표적 특질이라 할 것이다. 그러나 현대 테크놀로지와 관련된 제품의 복잡성 및 대단위 정보 디자인의 등장은 두번째 모드의 중요성을 일깨우고 있다. 즉, 단순한 조형적 아름다움보다는 사용성을 목표로 하는 효과적 커뮤니케이션의 문제가 인터랙션(Interaction)되는 각종 정보를 디자인하는데 가장 중요한 테마가 된다는 것이다.

본 논문은 디자이너의 두번째 모드인 논리, 체계성이 GUI 디자인 분야에서 인지적 시각 언어를 전개하는데 필요한 방법론의 기초가 되고 있음을 보여 준다. 이것은 디자이너가 인터랙션 원리들을 어떻게 시각언어에 적용하여 성공적으로 제품으로서 완성되게 할 것인가의 구체적 내용이 다.

산업혁명으로 제품디자이너가 태어나고 매스미디어의 발달로 디자이너가 광고디자인분야에서 창의력을 발휘해왔듯이 컴퓨터 테크놀로지가 대중화되면서 인터랙션디자인이라는 새로운 디자인 분야가 디자인 커뮤니티안에 자리잡기 시작하고 있다. 그러나 인터랙션 디자인이 갖고 있는 제품적 의미를 얼마나 진지하게 이슈화되고 있는가는 의문이다. 사실, 긴밀한 연관을 갖고 있어야 하는 HCI(Human Computer Interaction)와의 관계 형성도 미미하다. 물론 미국에서도 HCI학회의 디자이너 참여는 2%밖에 되지 않는다고 한다. 그 이유는 아직도 시각언어가 일반의 언어처럼 과학화되지 못한데 있고 GUI디자이너가 효과적인 시각 커뮤니케이션, 즉 사용성을 목표로 해서 정보를 분석하고 시각언어로 종합하는 분석적 과학으로의 접근이 부족하기 때문이다. 현대 제품의우수함은 다양한 배경의 전문가들이 얼마나 완전한 커뮤니케이션을 이루어 제품에 반영하는가에 달

10. John Chris Jones, Design Methods, 1992.

려 있다. 이러한 개념이 현대조직이나 제품의 다양성과 복잡성을 극복하는 공동협력 학제간 팀웍(Collaboration and Interdisciplinary Team work)이다. 디자이너가 사용하는 시각언어에 대한 사용적 접근은 사용자와의 효과적인 커뮤니케이션 뿐 아니라 다른 배경의 전문가와의 효과적 커뮤니케이션으로 소프트웨어 개발에 있어 학제간 팀웍을 이끈다.

다양화되는 복잡한 정보의 홍수 속에서 효과적인 시각 커뮤니케이션의 창조는 점점 중요한 과제로 대두되고 있으며 이에 인터랙션 디자인의 의미와 방법론의 연구는 시각언어의 표현적 스킬(skill)에 우선되어야 한다.

참고문헌

1. Kevin Mullet & Darrell San, Designing Visual Interfaces, 1995, SunSoft Press
2. Virginia Howlett, Visual Interface design for Windows, 1996, John Wily & Sons, Inc
3. Jakob Nielsen, Usability Engineering 1993, Academic Press, Inc.
4. Aaron Marcus, Screen Design for Electronic Document 1992 ACM Press
5. D. W. Hanlyn, The Psychology of Perception: A Philosophical Examination of Gestalt Theory and Derivative Theories of Perception (Studies in Philosophical Psychology), 1961
6. Terry Winograd, Bringing Design to Software 1996, ACM press
7. Brenda Laurel, The Art of Human Computer Interface Design 1990, Addison-Wesley Publishing Company
8. 윤철호, 인간 컴퓨터 인터페이스, 1996, 대영사
9. John Chris Jones, Design Methods 1992, Van Nostrand Reinhold